LA NUOVA RIVISTA INTERAMENTE DEDICATA AL MONDO LINUX

2 REGALI! MOUSEPAD + CD-ROM

www.oltrelinux.com

Linux&C.

MENSILE - ANNO 1 NUMERO 1 - NOVEMBRE L. 9.000 € 4,65

ISSN 1129-2296

90001

9 771129 229009

LINUX&C. - ANNO 1 - NUMERO 1 - DISTRIBUZIONE ME.PE. - PISCOPO EDITORE

NEL CD DI QUESTO NUMERO

- **WORD PERFECT 8**
  Il programma di Word Processing tra i più completi, potenti ed intuitivi.
- **INTERNET**
  Dal collegamento alla navigazione, dalla configurazione della posta elettronica ai newsgroup: tutto l'indispensabile perchè internet sia davvero facile.
- **DOCUMENTAZIONE**
  Una enorme raccolta di how-to, mini-howto, manuali per l'utente e lo sviluppatore. Una miniera di informazioni.
- **MASTERIZZAZIONE**
  Tutta una serie di programmi, grafici e non, per masterizzare senza difficoltà anche da Linux.
- **SVILUPPO**
  Compilatore C/C++, Pascal, JDK, Database, PHP, Security, Programmazione sotto Gnome

RECENSIONE

Corel WordPerfect 8

Un facile e potente WordProcessor, perchè non si pensi che Linux sia sprovvisto di strumenti di produttività. Ed è solo l'inizio...

HACKERS-MANIA

Difendiamoci con i firewall

I firewall sono lo strumento di sicurezza più utilizzato in assoluto. Cerchiamo di capire cosa sono, come si installano e come si configurano.

CONFIGURAZIONE

INTERNET facile

Configurare l'accesso ad internet può presentare qualche difficoltà. Una guida per i nuovi utenti.

LINUX A SCUOLA

NetQuiz: un progetto «scolastico»

Continua lo sviluppo del software per la realizzazione di Quiz. Un progetto client-server su piattaforma mista Linux/Windows.

BeOS: un nuovo sistema operativo dalle caratteristiche interessanti

Linux

Se non ci fosse bisognerebbe inventarlo! Colombo

ED ANCORA...

- **INSECURITY NEWS**
  Notizie di bugs&patches.
- **INSTALLAZIONE**
  Questa volta si parla di FreeBSD.
- **PROGRAMMAZIONE**
  Socket: cosa sono, come si usano, a cosa servono...
- **LINUX C/C++**
  Continuiamo con le shell grafiche e non disponibili per il Gcc: rimarrete stupiti.
- **LATEX**
  Continuiamo il corso di Tex, per comporre testi scientifici facilmente.
- **...ED ALTRO ANCORA!**
  XML - il futuro dell'html - X-Window
  Sito Web del mese.
  Un salto in libreria...

SMAU, successo annunciato...

Sembrava fatta. Per mesi abbiamo immaginato come doveva essere la rivista. Per mesi abbiamo provato a intuire quale fossero le migliori rubriche, quelle più essenziali ma allo stesso tempo accattivanti e in grado di rispecchiare appieno il gusto di un pubblico che decisamente ha molte caratteristiche ma, in primis (pensiamo), quella di non accontentarsi delle solite trite e ritrite "telenovele informatiche". Un pubblico che non giudica la bontà di un giornale dal numero di pagine ma dal numero di pagine leggibili. Per mesi siamo stati consci del fatto che non si possa basare un mensile di informatica esclusivamente sui confronti tra le proprietà tecniche, economiche se non addirittura "terapeutiche" di un processore o di una stampante. Era stata un'impresa impegnativa ma, dopo centinaia di mega salvati, cancellati nel tentativo di raggiungere l'ottimo (o quello che noi riteniamo tale) sembrava che tutto finalmente giungesse al termine. Le scelte di impaginazione, quelle dei font e dei colori, i tagli e le aggiunte ai vari articoli. Tutto pareva essere "settato" nella maniera migliore. Sembrava davvero fatta. Eppure ad un certo punto (cioè oggi, 4 ottobre, al ritorno dall'ultimo giorno di Smau) mi sono accorto che il peggio doveva ancora venire.

Già. L'editoriale. In anni di lettura di riviste specializzate, infatti, non avevo mai lesinato in due cose: i decimi di miopia (ovviamente persi) e le critiche nei confronti di coloro che redigevano editoriali. Questo è troppo banale, quest'altro è superato, quest'ultimo scritto da uno che si chiede dove acquistare le schede perforate, perchè non riesce a trovarle più nello store sotto casa.

Già. Il fatto è che ho sempre pensato che in un Editoriale vi sia un concentrato del meglio della "filosofia esistenziale" che dovrà caratterizzare il giornale per cui mi sono sempre aspettato un pezzo da antologia moderna, un sunto di capacità tecniche e letterarie.

Già. E adesso? Mantengo uno stile asciutto e tecnico oppure scrivo a "cuore aperto" per esprimere le mie gioie, le mie paure e le mie aspettative?

Già. Il problema vero è che non mi ero mai reso conto perfettamente di come fosse difficile riuscire a soddisfare in alcune righe i tanti "perchè" che potenzialmente possono affollare la Ram di chi ha deciso di comprarti il primo numero.

Il lettore, infatti, ti offre, con quel gesto, una buona dose di credito, una sorta di cambiale in bianco... ma con una precisa scadenza (una specie di trial cartaceo). Sei tu che devi rispondere via via ai vari perchè egli debba tornare nuovamente a darti fiducia in quella selva di proposte che si ritrova nell'oscuro gabbiotto dell' edicolante dove, spesso, le cose più illuminate sono tali per mancanza di luce propria.

Ecco, è questo che dovrebbe riuscire a fare un Editoriale: dare delle risposte. Ma i perchè sono davvero tanti e le risposte ancor di più. Soprattutto in un mondo in cui molto, se non tutto, deve ancora essere scoperto. Per questi motivi posso fare unicamente una cosa: darvi appuntamento al mese prossimo. L'Editoriale del numero 1 termina così, con un rimando... Altro non aggiungo, se non i doverosi ringraziamenti a Suse e RedHat, che hanno distribuito ai relativi stand le copie omaggio e a tutti gli altri che hanno contribuito alla diffusione del numero 0.

Linux & C.

Mensile di Informatica con supporto integrativo allegato

Novembre 1999 - Anno 1 Numero 1

Editore
Piscopo Editore s.r.l.
Via Ruffini, 2/a
00195 Roma

Direttore Responsabile
Annabella Faustini
(a.faustini@oltrelinux.com)

Direttore Editoriale
Patrizio Tassone
(p.tassone@oltrelinux.com)

Grafica ed Impaginazione
Edizioni Vinco
grafica@oltrelinux.com

Hanno Collaborato:
Emanuele Busuito
(e.busuito@oltrelinux.com)
Francesco Consumi
(f.consumi@oltrelinux.com)
Marco Ivaldi
(m.ivaldi@oltrelinux.com)
Jilani Khaldi
(j.khaldi@oltrelinux.com)
Lorenzo Lazzeri
(l.lazzeri@oltrelinux.com)
Felice Mainolfi
(f.mainolfi@oltrelinux.com)
Riccardo Murri
(r.murri@oltrelinux.com)
Silvia Sammarro
s.sammarro@oltrelinux.com
Salvatore Sanfilippo
(s.sanfilippo@oltrelinux.com)
Filippo Stefanelli
(f.stefanelli@oltrelinux.com)
Ivan Trevisan
i.trevisan@oltrelinux.com

Pubblicita
Top Media
Tel. 02 / 26.68.05.47 r.a.
Fax. 02 / 26.64.816
(pubblicita@oltrelinux.com)

per contattare la redazione
Edizioni Vinco
Redazione Linux & C.
Casella Postale 41
51010 Massa e Cozzile
Pistoia
Tel. 0572 - 91.39.46
Fax. 0572 - 90.10.02
redazione@oltrelinux.com

Stampa
Sofimel s.r.l. - Torino

Fotolito
M&C Graphicpoint - Roma

Distribuzione
Me.Pe. S.p.A.
Via Giulio Carcano, 32 - Milano
Tel. 02 / 89.59.21

Numero registrazione al Tribunale di Roma: 332/99 del 17.07.1999



In questo numero...

SOMMARIO

Novembre 1999 Numero 1

Legenda Articoli

- 🐧 BASE
- 🐧🐧 INTERMEDIO
- 🐧🐧🐧 AVANZATO
- 🐧🐧🐧🐧 EXPERT

Il numero 0 della rivista è sul cd-rom in formato pdf!





Nel Cd-Rom del numero 1

Il software sul CD

Presenteremo qui il software incluso nel CD-ROM allegato, con brevi note per descrivere cos'è e com si installa.

Innanzitutto, per accedere al CD-ROM è necessario che quest'ultimo sia montato, cosicché il suo contenuto possa essere acceduto da Linux.
Per montare il CD-ROM è necessario procedere come segue, da utente root:

```
mount /dev/cdrom /mnt/cdrom
```

Dove /dev/cdrom è un link simbolico al device attraverso il quale si accede al lettore di CD-ROM, mentre /mnt/cdrom è la directory che funge da mountpoint per il filesystem.

Acrobat Reader 4.0

Per scompattare e installare il noto programma di Adobe per la visualizzazione di file in formato Postscript e PDF (Portable Document Format), seguire le seguenti istruzioni; in una directory scompattare il pacchetto presente sul CD-ROM con il comando tar:

```
tar xvfz /mnt/cdrom/pspdf/
                    linux-ar-40.tar.gz
```

In questo modo verrà creata la sottodirectory ILINXR.install entro la quale sarà presente il file INSTALL da eseguire per procedere all'installaizione vera e propria,
durante la qualei verrà chiesto in quale directory intendiamo installare il programma: suggeriamo /opt/Acroread4.
Per lanciare il Acrobat Reader 4, sarà in seguito sufficiente eseguire il programma:

```
/opt/Acroread4/bin/acroread
```

The Gimp

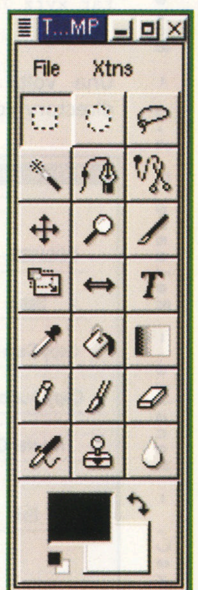
Il famoso programma di fotoritocco e disegno.
Per installarlo è possibile utilizzare i binari in formato RPM, per la distribuzione RedHat, o DEB, per la distribuzione debian, o compilarsi in prorpio l'eseguibile, procedura che puo' essere effettuata su qualsiasi distribuzione.
In quest'ultimo caso, procedere come segue:

```
tar xvfz /mnt/cdrom/graphics/
                    gimp-1.1.9.tar.gz
```

Per scompattare i sorgenti. Per compilare il tutto e installarlo i comandi necessari sono i seguenti:

```
cd ./gimp-1.1.9
./configure
make
make install
```

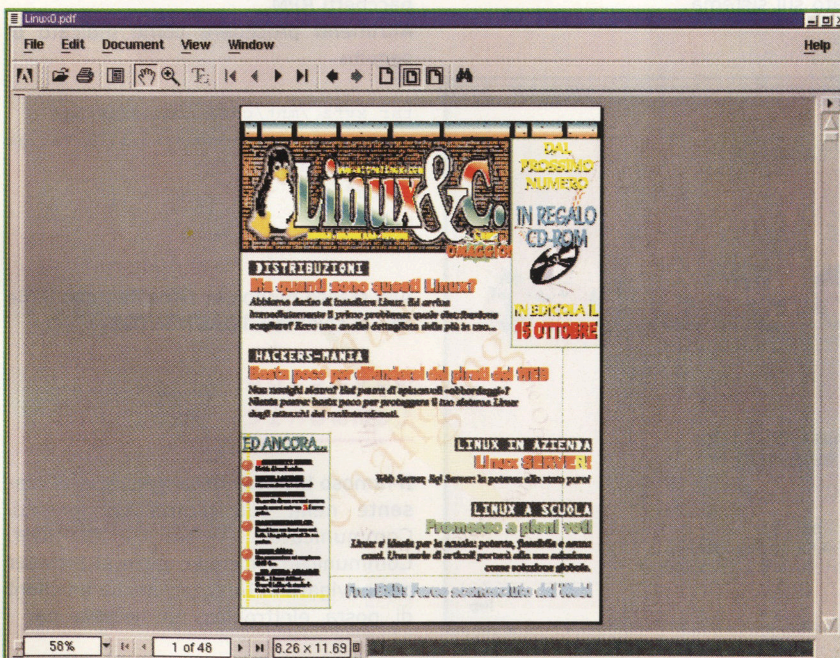
The Gimp è un programma grafico che, a benché sia completamente libero come licenza d'uso, non ha niente da invidiare a prodotti commerciali più conosciuti.



Il Kernel

Per installare il kernel 2.2.12, l'ultima versione stabile al momento della creazione del CD-ROM, si consiglia di procedere come segue.

Innanzitutto è bene creare una directory chiamata linux-2.2.12 all'interno di /usr/src:



```
mkdir /usr/src/linux-2.2.12
```

In seguito creare un link a tale directory, chiamato linux sempre all'interno di /usr/src:

```
ln -s /usr/src/linux-2.2.12
  /usr/src/linux
```

Entrare in /usr/src/ e scompattare il pacchetto contenente i sorgenti del nuovo kernel:

```
cd /usr/src
tar xvfz /mnt/cdrom/kernel/
                         linux-2.2.12.tar.gz
```

A questo punto in /usr/src/linux ci troveremo i sorgenti dell'ultimo kernel. Per procedere alla compilazione dello stesso sarà dunque necessario avviare la configurazione. Fare riferimento al Kernel-HOWTO per maggiori dettagli.

XCDroast

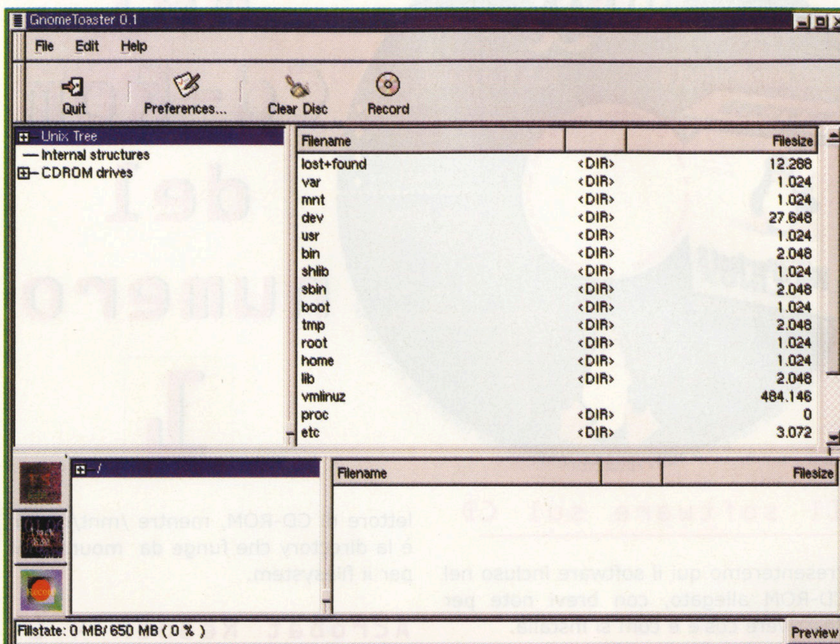
L'utile programma XCDroast per la masterizzazione di CD tramite l'interfaccia grafica, si installa scompattando il pacchetto presente sul CD-ROM e compilandone i sorgenti:

```
tar xvfz /mnt/cdrom/cdmastering/
                    xcdroast-0.96e.tar.gz
```

Una volta scompattato entrare nella directory xcdroast-0.96e.tar.gz e compilare con e soliti comandi:

```
cd ./xcdroast-0.96e
./configure
make
make install
```

XCDroast è un frontend per cdrecord, l'interfaccia grafica cioé per un programma a linea di comando già esistente. Necessita perciò che cdrecord sia installato sul sistema.



Gcombust/Gtoaster

Gcombust e Gtoaster sono altri frontend per cdrecord, scritto appositamente per funzionare all'interno di GNOME. Il loro stato di sviluppo è un po' arretrato rispetto a XCDroast, ma vale sicuramente la pena provarli.

Per installarli, se si sta usando su una RedHat è consigliato comunque appoggiarsi ai binari precompilati contenuti nei pacchetti RPM.

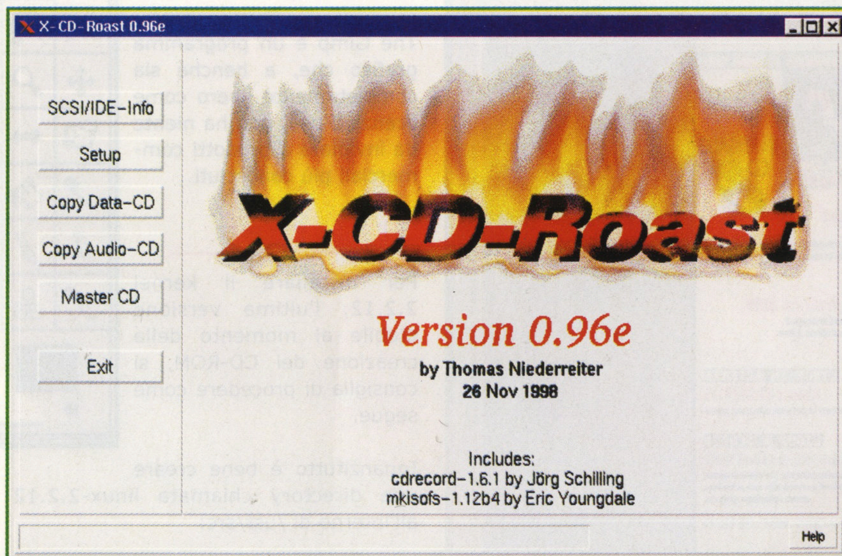
Altrimenti procedere come indicato di seguito

```
tar xvfz /mnt/cdrom/cdmastering/
                    gcombust-0.1.22.tar.gz
cd gcombust-0.1.22
./configure
make
make install
```

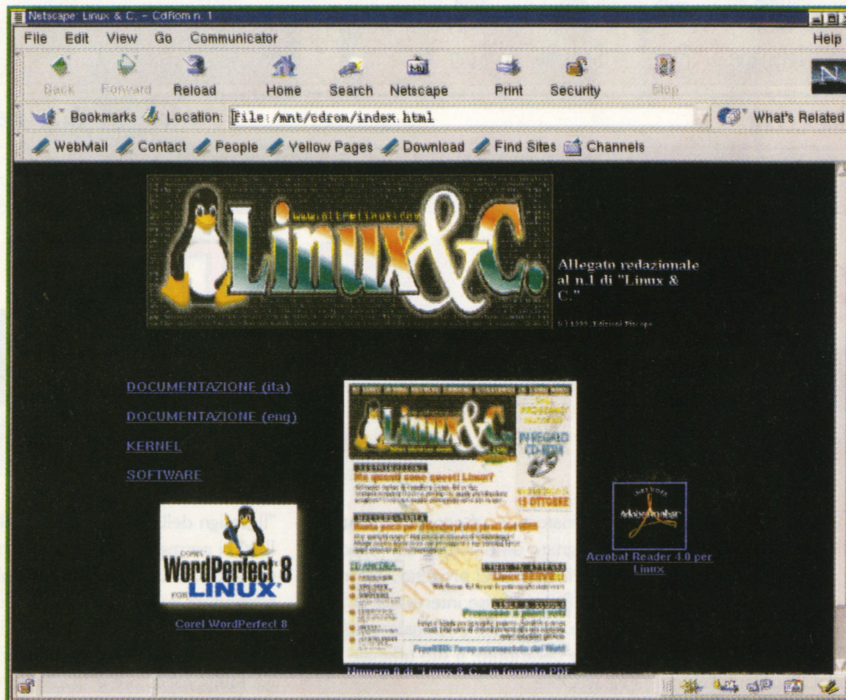
Allo stesso modo per Gtoaster, con il file che si chiama GTOASTER.TGZ

Netscape Communicator 4.61

Il famoso browser della Netscape, è presente nella sua ultima versione di Communicator sul CD-ROM. Communicator include, oltre al browser per navigare in Internet, anche un client di posta elettronica, un lettore per i







newsgroup, un editor visuale per pagine web.

Per l'installazione e' necessario scompattare il pacchetto contenente i binari del programma e procedere successivamente all'installazione vera e propria:

```
tar xvfz /mnt/cdrom/internet/
         browser/communicator-
         v461-export.x86-unknown-
         linuxglibc2.0.tar.gz
cd ./communicator-v461.x86-
         unknown-linux2.0
./ns-install
```

L'ultimo comando eseguito lancia lo script di installazione che si occuperà di creare le directory giuste e scompattare ulteriormente i vari file.

Gnu C Compiler

Sul CD-ROM è presente una directory contenente tutti i componenti del compilatore C della GNU, con i vari tool di cooredo (compilatori C++, fortran, Java e Objective C, le binutils, autoconf, eccetera).

Per installare i pacchetti è necessario procedere sistematicamente su ognuno di essi scompattandoli in una directory, compilandone i sorgenti e installando.

Ad esempio, per il pacchetto gcc-2.95.1:

```
tar xvfz /mnt/cdrom/develop/
         gcc/gcc-2.95.1.tar.gz
cd gcc-2.95.1
./configure
make
make install
```

E' comunque consigliato, prima di procedere alla compilazione con il comando make, leggere attentamente i vari file di documentazione presenti nella sottodirectory, al fine di stabilire con certezza quali opzioni di compilazione ci sono necessarie.

Gnome

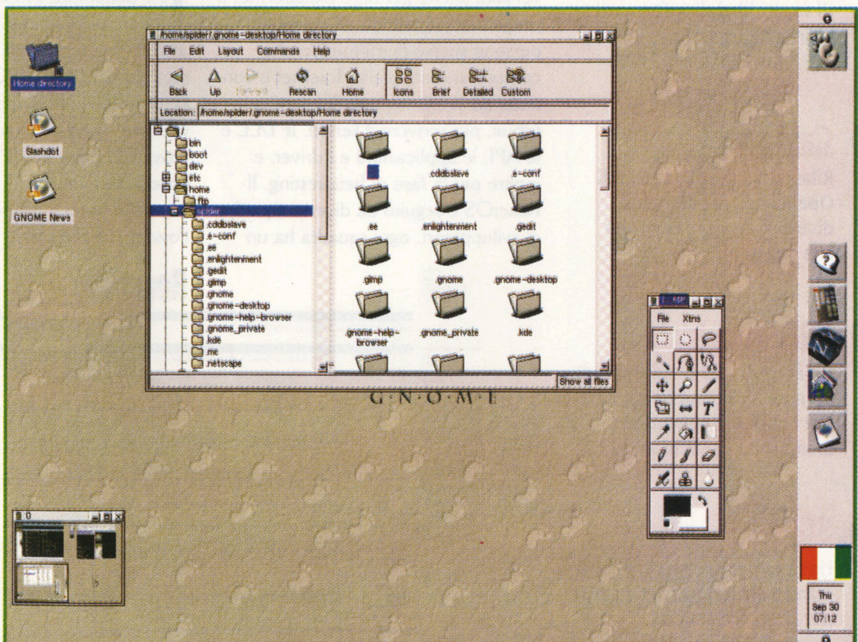
E' presente sul CD-ROM una directory contenente i sorgenti del sistema base GNOME.

Per procedere alla compilazione e installazione è necessario procedere come spiegato per il compilatore GNU C compiler.

Esempio:

```
tar xvfz /mnt/cdrom/gnome/
         gnome-libs-1.0.17.tar.gz
cd gnome-libs-1.0.17
./configure
make
make install
```





File Edit View Go Window Help

 Back  Forward  Reload  Home  Search  Netscape  Print  Security  Stop

 Bookmarks  Location: http://www.oltrelinux.com  What's Related

NEWS UPDATE

a cura di Marco Ivaldi (raptor)



ReactOS

Un nuovo sistema operativo opensource compatibile con i drivers e applicazioni WindowsNT.



Total Impact

Sistema Multiprocessore (da 5 a 13 CPU) sfruttando motherboard OpenPlatform di IBM.



Progetto FreeVMS

Una versione di VMS completamente free, che potrà girare anche su piattaforme x86.



Lutto nel mondo Unix

W. Richard Stevens, autore di molti libri di informatica e programmazione, è deceduto.



Caldera

Rilasciata la versione 2.3 di OpenLinux, la prestigiosa distribuzione Linux rivolta alle aziende.



ISI&Mitsubishi

Un accordo per lo sviluppo di un nuovo sistema operativo per palm-top: concorrenza annunciata per WIndowsCE?



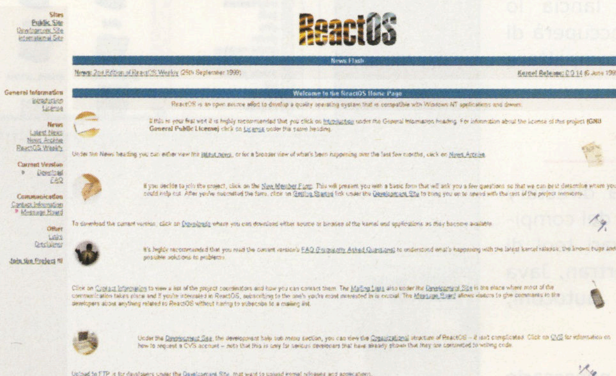
SGI

Un nuovo progetto di standardizzazione di OpenGL per Linux, cercando la compatibilità con Mesa.

ReactOS: un kernel open source compatibile con Windows NT

Il progetto ReactOS si pone l'obiettivo di creare e mantenere un sistema operativo open source di qualità, compatibile con i driver e le applicazioni di Windows NT esistenti. Per ora il kernel sviluppato è giunto alla versione 0.0.14 del 6 Giugno 1999, ed è solamente per i sistemi compatibili Intel, ma in futuro non è da escludere che possano essere introdotte versioni anche per altre piattaforme. Il progetto è portato avanti da molte persone sparse per il mondo che comunicano attraverso Internet e continua ad avere bisogno di nuovi contributi, per scrivere il kernel, le DLL e le API, le applicazioni e i driver, e inoltre per la fase di beta-testing. Il ReactOS è seguito da diverse squadre di sviluppatori; ogni squadra ha un coordinatore, e ad ogni coordinatore è assegnato un obiettivo dal responsabile principale del progetto, Jason Filby. Per chi fosse interessato al progetto e intendesse eventualmente dare un suo contributo, il sito ufficiale del ReactOS è

http://www.reactos.com



Sistema Linux PowerPC Multiprocessore

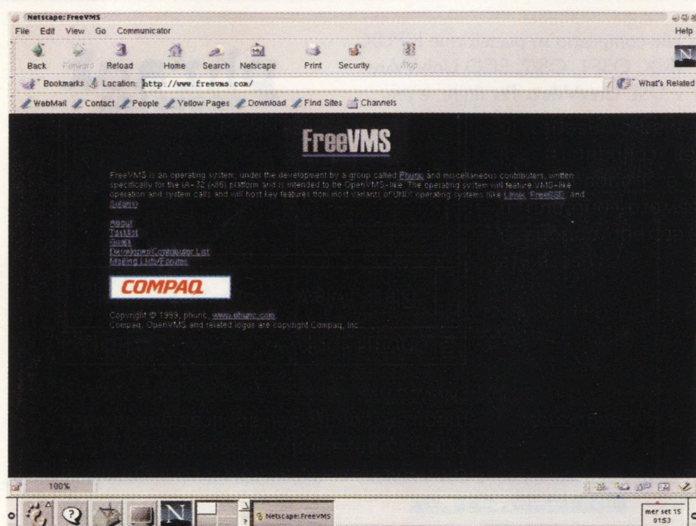
Il 1 Settembre 1999 la Total Impact ha annunciato che intende integrare la sua tecnologia TotalMP con la motherboard PowerPC Open Platform della IBM, per creare il più versatile sistema Linux basato su PowerPC esistente. Il TotalMP Linux system permetterà di utilizzare da 5 (modello base) a 13 processori PowerPC nello stesso sistema desktop. "Il design della motherboard della IBM ci permette di integrare la nostra tecnologia multiprocessore e inoltre di supportare il processore G4", dice Brad Nizdil, il presidente della Total Impact. "Siamo esperti nella tecnologia multiprocessore e stiamo correntemente utilizzando un sistema a 128 processori sotto MacOS". Il sistema dovrebbe essere disponibile entro Aprile del prossimo anno. I prezzi ancora non sono stati determinati.

Il progetto FreeVMS: VMS su Intel e compatibili

FreeVMS è un sistema operativo correntemente in realizzazione scritto specificatamente per le piattaforme x86: l'obiettivo è creare un OS simile al VMS, ma con caratteristiche tipiche di alcuni sistemi Unix (come Linux, FreeBSD e Solaris). Gli sviluppatori fanno parte di un gruppo chiamato Phunc (http://www.phunc.com), aiutati da numerosi volontari esterni. OpenVMS non è mai stato sviluppato per funzionare su architetture diverse da VAX o Alpha, nonostante esso sia incredibilmente potente rispetto ad altri sistemi operativi. Molte persone non possono perciò sperimentare un OS così interessante, per via del fatto che esso richiede piattaforme generalmente troppo costose. Così alcuni membri del Phunc hanno deciso di cominciare a sviluppare questa versione Open Source per Intel: il progetto è partito ufficialmente il 3 marzo







1999. Per chiunque fosse interessato a partecipare al FreeVMS, basta semplicemente iscriversi alla mailing list del progetto e comunicare le proprie competenze. Sito ufficiale:

http://www.freevms.com

Una perdita nel mondo Unix

W. Richard Stevens, noto autore di libri sull'informatica e la programmazione, è deceduto il 1 Settembre. I suoi libri "UNIX Network Programming", "Advanced Programming in the UNIX Environment" e "TCP/IP Illustrated" (considerata la "bibbia" del TCP/IP) sono delle pietre miliari per i programmatori e gli sviluppatori Unix di tutto il mondo. Nato nel 1951, Stevens era originario di Luanshya, nella Rhodesia del Nord (ora Zambia). Successivamente la famiglia si spostò in numerose città americane, da Salt Lake City a Washington, fino a stabilirsi in Sudafrica. Richard seguì poi la Fishburne Military School a Waynesboro, in Virginia. In seguito ricevette una laurea in Ingenieria Aerospaziale dall'Università del Michigan e in Ingegneria dei Sistemi dall'Università dell'Arizona. Si trasferì a Tucson nel 1975 e fu impiegato al Kitt Peak National Observatory come programmatore fino al 1982. Dal 1982 al 1990 è stato Vice Presidente del Computing Services presso l'Health Systems International a New Haven, CT, tornando poi a Tucson nel 1990. Qui seguì la sua carriera di autore e consulente.

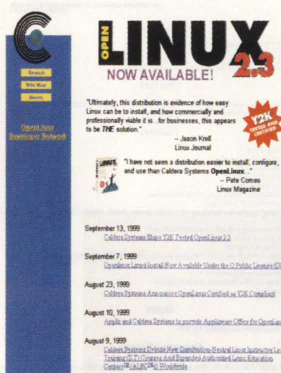
Caldera OpenLinux 2.3

È uscito ai primi di settembre OpenLinux 2.3 della Caldera. OpenLinux è una distribuzione Linux completamente testata, stabile e molto curata, ideale sia per uso casalingo che per le aziende. I suoi punti di forza sono l'affidabilità, la semplicità di installazione e amministrazione, la sicurezza e la presenza di applicazioni robuste. Inoltre Caldera Openlinux 2.3 include l'ultima versione di LIZARD (Linux Wizard), un potente ma facile da usare tool di installazione grafico. Esso realizza automaticamente il detect dell'hardware (mouse, tastiera, video, schede sonore, CD-ROM, dischi fissi, floppy, schede PCMCIA ed ethernet...) per rendere OpenLinux utilizzabile subito, senza preoccuparsi troppo della configurazione. OpenLinux include inoltre PartitionMagic e BootMagic, per condividere lo stesso PC con Windows. Nel pacchetto troverete anche il K Desktop Envirinment (KDE), la office suite Applixware 4.4.2, il nuovo Netscape Communicator 4.61, StarOffice 5.1 in sei lingue e molto altro ancora. Il sito ufficiale della Caldera e'

http://www.calderasystems.com/

ISI e Mitsubishi: un nuovo sistema operativo?

La Integrated Systems Inc., una importante software house americana, svilupperà a nuovo sistema operativo per computer portatili con la giapponese Mitsubishi. Così hanno annunciato lunedì 6 Settembre i responsabili delle due compagnie. La Integrated Systems è il più grande produttore del mondo di sistemi operativi per uso con microcontroller avanzati. Il suo portavoce ha spiegato che le due compagnie intendono sviluppare un sistema simile agli attuali palm-top correntemente sul mercato, che attualmente utilizzano Windows CE della Microsoft come sistema operativo. I nuovi gadget avranno un accesso internet ad alta velocità, oltre a un word processing migliorato e funzioni di agenda e rubrica. Pare inoltre che la ISI stia considerando di cooperare con altre tre società elettroniche giapponesi, per lo stesso progetto: molto probabilmente la Hitachi, la NEC e la Fujitsu.





SGI e Mesa insieme per standardizzare Linux/OpenGL

SGI ha annunciato il suo nuovo progetto di standardizzare OpenGL su Linux, tenendo però conto della compatibilità con Mesa. Una tappa importante nel processo di affermazione di Linux, tenendo conto che si tratta della standardizzazione di una delle tecnologie più importanti della SGI. I partecipanti questo progetto includono persone che fanno parte di: Linux Standard Base, Mesa, Metro Link, NVIDIA, Parametric Technology Corporation, Precision Insight, SGI, XFree86 e Xi Graphics. Per ulteriori informazioni:

File Edit View Go Window Help



Bookmarks Location: http://www.oltrelinux.com What's Related

GnuPG
Crittografia OpenSource per Linux: un gratuito sostituto di PGP, interamente compatibile RFC2440 (OpenPGP).

Phrack 55
Il magazine underground più famoso ha ormai raggiunto quota 55, annoverando contributi di tutti rispetto.

KDE
Uno dei migliori desktop per linux (se non addirittura il migliore), giunto alla versione 1.1.2.

Linux: Kernel 2.4
Linus Torvalds annuncia per fine anno il nuovo kernel 2.4.x, con la tanto discussa opzione "feature freeze".

Sun MicroSystems
Sun annuncia i nuovi SunRay1, thin-client di ridotte dimensioni e caratteristiche molto interessanti.

FreeBSD 3.3
La nuova versione del S.O., giunto ormai alla versione 3.3, attesa dallo scorso maggio.

Happy Birthday, Linux!

Packet Storm
Dopo la chiusura improvvisa di qualche mese fa, il famoso sito sulla sicurezza torna a vivere...

http://www.sgi.com

GnuPg 1.0 finalmente disponibile

GnuPG 1.0.0 è uscito il 7 settembre 1999, dopo una lunga attesa. GnuPG è un completo e gratuito sostituto al famoso programma di crittografia PGP. Non utilizzando IDEA o RSA, può essere usato senza alcuna restrizione anche al di fuori degli Stati Uniti, non essendo soggetto alle leggi sull'esportazione. GnuPG è interamente compatibile con l'RFC2440 (OpenPGP). Esso è completamente gratuito e soggetto alla licenza GPL ed è stato scritto totalmente da zero. Le possibilità sono innumerevoli: può infatti essere utilizzato come filtro, è in grado di decriptare e verificare i messaggi PGP 5.x, ha alcune migliorie rispetto a PGP stesso, supporta gli algoritmi ElGamal, DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 e TIGER. È inoltre possibile utilizzare dei moduli di estensione per implementare semplicemente nuovi algoritmi. Tutto questo e molto altro... I sistemi supportati sono numerosi. Primo fra tutti Linux, su x86, alpha, sparc64, m68k e powerpc, seguito da FreeBSD e OpenBSD. È inoltre possibile compilarlo su piattaforme HPUX, IRIX, OSF1, OS/2, SCO UnixWare e USL UnixWare. Per finire, è presente una versione per Windows 95/NT. Il sito ufficiale è

http://www.gnupg.org .

[Home] [Download] [What's New]
[Mirrors] [Documentation] [More Crypto] [Other Sites]

News Flashes

GNU PRIVACY GUARD 1.0 available now!

Overview

GnuPG is a complete and free replacement for PGP. Because it does not use IDEA or RSA it can be used without any restrictions. GnuPG is a RFC2440 (OpenPGP) compliant application.

Features

- Full replacement of PGP.
- Does not use any patented algorithms.
- GPLed, written from scratch.
- Can be used as a filter program.
- Full OpenPGP implementation.
- Better functionality than PGP and some security enhancements.
- Decrypts and verifies PGP 5.x messages.
- Supports ElGamal (signature and encryption), DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 and TIGER.
- Easy implementation of new algorithms using extension modules.
- User id is treated in a standard format.
- Support for a key expiration date.
- English, French, German, Italian, Polish, Portuguese (Brazilian), Russian and Spanish language support.
- Online help system.
- Optional anonymous message receivers.
- Integrated support for HKP keyservers (wwwkeys.pgp.net)
- and many more things...

Status

Version 1.0.0 has been released on September 7th, 1999.

Supported Systems

GnuPG works fine on GNU/Linux with x86, alpha, sparc64, m68k or powerpc CPUs. (x86 is my primary development system the other CPUs are only checked from time to time)

Phrack 55

La e-zine underground più famosa del mondo è giunta alla sua cinquantacinquesima uscita. Lo staff di Phrack e route annunciano la presenza sul sito ufficiale:

http://www.phrack.com/phrack55.tar.gz

La storia di Phrack affonda le sue radici nel mondo delle BBS amatoriali americane: la prima uscita, a cura degli storici Taran King e Knight Lightning, è datata 17 novembre 1985. Di li' in poi gli editors e gli autori si sono susseguiti, fino ad arrivare alla e-zine che possiamo ammirare oggi, che ha visto pubblicati articoli famosi come "smashing the stack" di Aleph One, il moderatore di bugtraq. In questo numero potete trovare, tra le altre cose: il tributo a W. Richard Stevens, il manuale delle Libnet, la sicurezza delle CGI in PERL, il Network Layer di Linux, uno studio sui buffer overflows di Win32, un'analisi sull'effettiva sicurezza dei firewall. Il sito ufficiale di phrack e'

phrack magazine

archive | search
contact us | submissions
forum | links

http://www.phrack.com

E' uscita KDE 1.1.2

Dal 13 Settembre è disponibile l'ultima versione dell'ormai famoso K Desktop Environment: KDE 1.1.2. I cambiamenti consistono principalmente in migliorie della veste grafica e in correzioni di bug presenti nelle precedenti versioni del prodotto. KDE è ormai il desktop standard per molte distribuzioni Linux, grazie alla stabilità e alla



9K read (at 1.2K/sec)



File Edit View Go Window Help

 Back  Forward  Reload  Home  Search  Netscape  Print  Security  Stop 

 Bookmarks  Location: http://www.oltrelinux.com  What's Related



potenza delle ultime versioni, come spiega Kurt Granroth, membro del team KDE. KDE 1.1.2 include il KDE Theme Manager, un nuovo tool per facilitare il cambiamento di tema del desktop, e supporta fino a 35 diverse lingue. KDE è disponibile per essere scaricata gratuitamente presso i numero mirror

http://www.kde.org/mirrors.html

oltre che dal server FTP primario

ftp://ftp.kde.org/pub/kde/stable/1.1.2/distribution/

Freezing per linux 2.4

Linus Torvarlds ha annunciato il 10 Settembre che per la fine dell'anno dovrebbe uscire il tanto atteso kernel 2.4.x. Un kernel che porta con se la tanto a lungo promessa 'feature freeze'. Questo non significa, spiega Linus, che i driver non verranno più aggiornati, nè implica che nuove parti di codice non possano essere aggiunte. Ad esempio anche driver (o filesystems) completamente nuovi possono venire accettati, a patto che non siano in conflitto con nulla e non implichino un approccio completamente nuovo a qualcosa. Un'altra tappa importante nello sviluppo di Linux.

Sun presenta i nuovi SunRay1

Sun Microsystems ha finalmente presentato i nuovi SunRay1, thin-client di nuova concezione, che rendono trasparenti agli utenti tutte le complessità operative, delegando la gestione della configurazione, dei drivers e quant'altro, alla sola postazioneserver. A differenza quindi dei tradizionali PC non esegue codice localmente. I vantaggi? Il più immediato riguarda l'aggiornamento del software, da effettuare ua volta sola sul server, per rendere disponibile tale modifica per tutti i SunRay1 collagati. L'unità base, grande non più di un libro, ha una dotazione standard di frame buffer 1280x1024 a 16milioni di colori, una scheda 10/100baseT, un lettore di SmartCard iso-7816, due porte USB, tastiera e mouse. Di sicuro interesse il fatto che con i SunRay1 non si è più legati ad una certa postazione: per lavorare su un altro client, basta sfilare la propria Card ed inserirla in un altro SunRay1, ritrovando il desktop come lo si era lasciato. Il prodotto è supportato da tutti i server Sun Enterprise e per i gruppi di lavoro che eseguono l'ambiente operativo Solaris, versione 2.6 o successive.

Uscito FreeBSD 3.3

È finalmente uscito FreeBSD 3.3, atteso ormai dallo scorso maggio. Nei mesi che sono intercorsi sono stati corretti numerosi bug e sono state apportate molte migliorie al già stabilissimo codice di questo Free OS. Tutti possono scaricarne gratuitamente una copia da:

ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/3.3-RELEASE/

Per informazioni su FreeBSD e sulla struttura della Release 3.3, fate riferimento al file ABOUT.TXT, mentre per le istruzioni di installazione leggete INSTALL.TXT e HARDWARE.TXT. Le novità sono molte. Particolarmente degne di nota le modifiche al kernel, che tra le altre cose portano una maggiore compatibilità per l'esecuzione dei file binari di Linux oltre che un miglioramento generale del bootstrap per gli i386. Sono stati inoltre aggiunti numerosi nuovi driver per gli hardware più disparati e corrette alcune debolezze per quanto riguarda la sicurezza. Per informazioni dettagliate su FreeBSD e per un elenco dell'hardware supportato:

http://www.freebsd.org

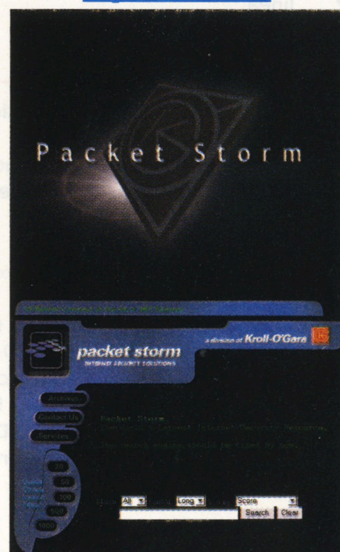
Buon compleanno, Linux!

La prima release pubblica ufficiale di Linux (versione 0.01) risale al 17 settembre 1991. Quello che forse e' oggi il più noto e controverso sistema operativo del mondo ha quindi ormai compiuto 8 anni. Buon compleanno, Linux!

Packet Storm e' tornato

Dopo essere stato chiuso alcuni mesi fa, e dopo essere stato al centro di una vicenda che ha coinvolto tutti gli appassionati di sicurezza di Internet, venerdì 22 settembre Packet Storm ha riaperto i battenti. Quello che è stato considerato da molti come il più completo archivio di software e informazioni sulla sicurezza informatica (e non solo) ha rivisto la luce grazie alla Kroll-O'Gara e al vecchio ideatore Ken Williams. Potete trovare il nuovo packet storm a

http://www.securify.com/packetstorm



 9K read (at 1.2K/sec)



Corel WordPerfect 8 For Linux

a cura di Lorenzo Lazzeri

Negli ultimi anni la diffusione di Linux è **aumentata** in modo sensibile. E con l'aumentare degli utenti di questo sistema operativo, è cresciuto anche l'interesse intorno a esso da parte di nuove e vecchie software house.

Corel, azienda canadese famosa per la sua suite Corel Draw in ambiente Windows, sta **investendo** molto su Linux, giungendo anche a sviluppare una **distribuzione**, in accordo con Debian e il team di KDE.

Parleremo qui di Corel Wordperfect 8, noto **wordprocessor** già esistente per altri sistemi operativi, di cui di recente è stata rilasciata una versione per Linux, la cui licenza è **gratuita** per usi **non** commerciali.

Dove trovarlo

E' possibile scaricare il file di installazione di Corel Wordperfect 8 dal sito http://linux.corel.com; il pacchetto ha dimensioni di circa 23 Mb.

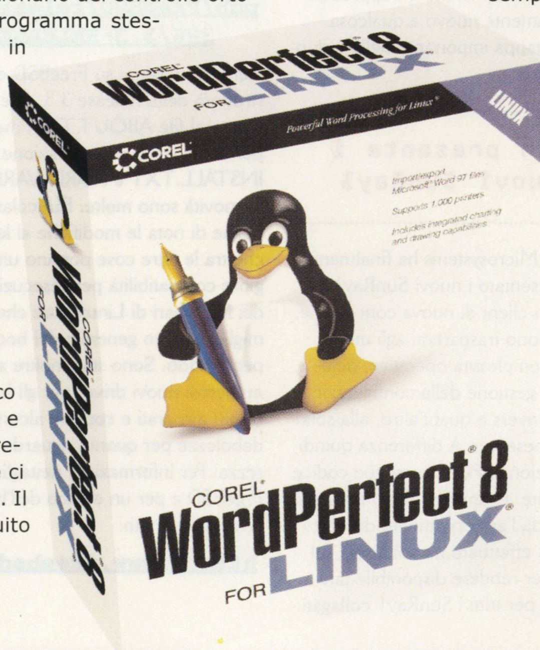
E' comunque presente sul CD allegato, nella directory /office/wp, che useremo come punto di riferimento per la fa se di installazione. Sul CD-ROM sono presenti anche i moduli per l'italiano, sia per quanto riguarda il dizionario del controllo ortografico, sia per quanto riguarda il programma stesso: voci di menu, pulsanti, help in linea.

Installazione

Per procedere all'installazione è necessario aver **montato** il CD-ROM, così da consentirci di accedere al suo contenuto. Procedere quindi come segue:

```
mount /dev/cdrom /mnt/cdrom
```

Dove /dev/cdrom è un link simbolico al device del lettore CD-ROM, e /mnt/cdrom è una directory che useremo come mountpoint, entro la quale ci troveremo il contenuto del CD stesso. Il comando mount dovrà essere eseguito da utente **root**.



Nella directory /tmp creiamo una sottodirectory e chiamiamola /tmp/wp8, ed entriamoci. Procediamo quindi a scompattare il programma e i suoi moduli per le lingue inglese e italiana:

```
continuity:/tmp/wp8$ gunzip
        </mnt/cdrom/office/wp/GUILG/GUILG00.GZ | tar xv

continuity:/tmp/wp8$ gunzip
      </mnt/cdrom/office/wp/GUILGIT/GUILGIT0.GZ | tar xv

continuity:/tmp/wp8$ gunzip
      </mnt/cdrom/office/wp/GUILGUK/GUILGUK0.GZ | tar xv
```

Notare che i comandi dovranno essere scritti su una sola riga, senza andare a capo.

Con questi tre comandi abbiamo quindi scompattato tutto il necessario per procedere all'installazione.

Esistono due tipi di installazione: in modo **testo** e in modalità **grafica**. Sarà il programma stesso a decidere in quale modalità funzionare: se verrà lanciato da dentro X partirà l'installazione grafica, che rimane comunque consigliata in quanto semplifica le varie fasi.

Lanciare quindi X e aprire una finestra di terminale, dove, da utente root (utilizzare il comando **su** per assumerne i privilegi se siamo soliti entrare nell'interfaccia grafica da utente normale - una buona abitudine) entreremo nella directory /tmp/wp8 e provvederemo a lanciare il comando

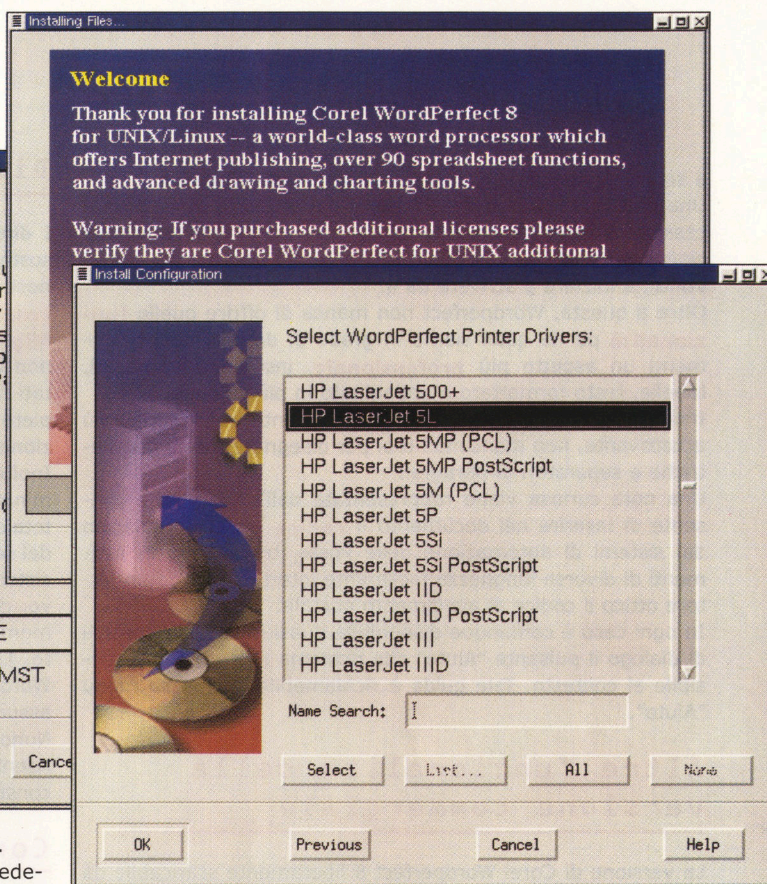
```
continuity:/tmp/wp8$ ./Runme
```

Il programma ci chede se abbiamo scompattato tutto, domanda alla quale possiamo rispondere tranquillamente in modo affermativo. Se la nostra **versione** del **kernel** non è tra quelle su cui alla Corel hanno effettuato i test, il programma ci avvertirà che tale versione **non è certificata**, chiedendoci se

In primo piano...







vogliamo continuare ugualmente. Possiamo rispondere sì.
A questo punto si aprirà una schermata grafica, che ci chiederà in che lingua mostrarci il contratto di licenza per il software che stiamo installando.
Proseguendo nell'installazione ci verrà chiesto in quale directory vogliamo installare Wordperfect. Consigliamo, per rispettare lo **standard** che si occupa di normalizzare l'uso del filesystem, di indicare /opt/wp8.
Dopodiché il programma ci chiederà se vogliamo effettuare un'installazione completa (Full) o di altro tipo e in seguito se vogliamo aggiornare una versione esistente del prodotto, chiedendoci di indicare la directory ove questo è attualmente installato. Se si tratta della **prima installazione** si consiglia di lasciare in **bianco** la risposta.
L'installazione proseguirà quindi facendoci scegliere quali moduli linguistici vogliamo adottare, le scelte possibili con i file che abbiamo usato sono l'italiano e l'inglese, anche insieme. Sempre per quanto riguarda la lingua, ci verrà chiesto quale vogliamo tra quelle scelte che sia quella predefinita.
Manca a questo punto solo di indicare il **tipo di stampante** in nostro possesso, e se vogliamo installare le icone per l'interfaccia grafica CDE.
Dopo l'ultima conferma, non resta che attendere che il programma di installazione proceda alla copia del programma.

Usare Wordperfect

Per lanciare Corel Wordperfect 8, il comando da eseguire è il seguente:

```
/opt/wp8/wpbin/xwp
```

È possibile, e soprattutto comodo, **aggiungere una voce di menu** o **un'icona** nel nostro ambiente grafico preferito, così da averlo sempre a portata di mano.
Una volta eseguito, Wordperfect ci chiederà di immettere il numero di licenza, necessario a **sbloccare** il limite di utilizzo del prodotto stesso, fissato in 90 giorni.
Per ottenere il numero di licenza è necessario registrarsi presso http://linux.corel.com
E' comunque possibile dire a Wordperfect che abbiamo intenzione di **rimandare** tale immissione e di usare quindi il wordprocessor.
Appena lanciato, Corel Wordperfect 8, ci presenta due finestre: una, più piccola, che consente di accedere alle **preferenze** del programma stesso, di aprire un'altra finestra per un nuovo documento, ecc.; l'altra, più grande e dall'aspetto familiare, almeno per chi è abituato a utilizzare questo genere di programmi, presenta un documento vuoto e i classici menu e toolbar, in italiano se questa è la lingua che abbiamo scelto di usare come predefinita.
L'interfaccia del programma, **ampiamente collaudata** dai tanti software di wordprocessing, è di utilizzo intuitivo.
Le peculiarità di Wordperfect risiedono soprattutto nella possibilità di **importare** ed **esportare** file in diversi tipi di formati, sia per quanto riguarda i testi sia per quanto riguarda tabelle e immagini.
È infatti possibile leggere e scrivere file nei formati delle diverse versioni di Microsoft Word, tra cui non manca Word97, Lotus Ami, le precedenti versioni di Wordperfect e tanti altri.
È in grado di importare anche tabelle nei formati Lotus 123, immagini Jpeg, GIF e tanti altri.
Questa caratteristica consente di poter **iniziare a prendere in considerazione** Linux quale unico sistema operativo per la propria computer, anche nel diffuso caso in cui si sia costretti

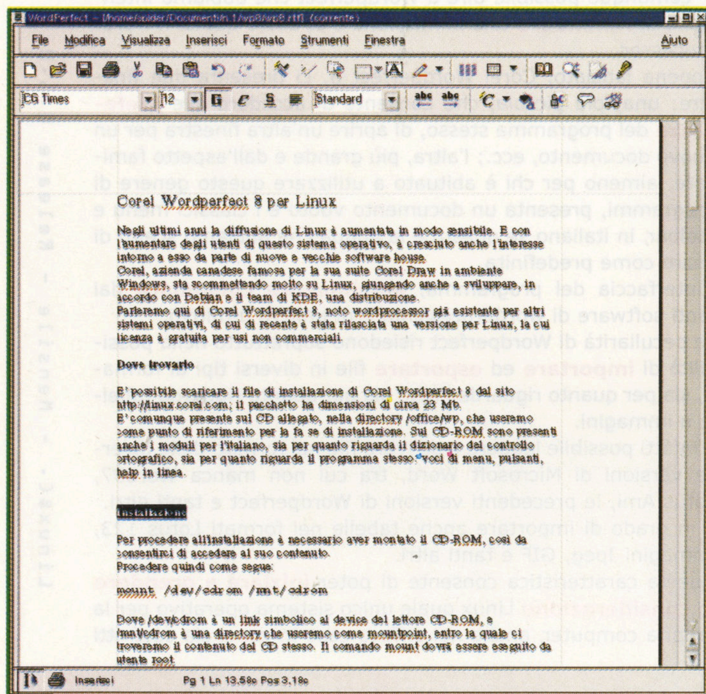


In primo piano...

a scambiare documenti con utenti di sistemi Windows.
Una novità nel campo delle feature dei software di word processing consiste nel cosiddetto **cursore ombra**, la possibilità cioè di poter clickare in qualsiasi punto del documento, anche vuoto, e iniziare a scrivere da lì.
Oltre a questa, Wordperfect non manca di offrire quelle **funzionalità** per le quali siamo in grado di dare ai nostri doumenti un aspetto più **professionale**, inserendo immagini, tabelle, testo formattato e impaginato in più colonne.
Inoltre, per abbellire il proprio documento rendendolo più accattivante, non mancano i tool per disegnare forme geometriche e separatori di paragrafi.
Una nota curiosa viene rappresentata dall'opzione che consente di inserire nel documento il **codice a barre** utilizzato dai sistemi di automazione delle Poste, traducendo in segmenti di diversa lunghezza facilmente interpretabili da un lettore ottico il codice di avviamento postale.
In ogni caso è comunque disponibile quasi in tutte le finestre di dialogo il pulsante "Aiuto" che richiama l'help in linea sensibile al contesto. Tale guida è richiamabile anche dal menu "Aiuto".

Altre funzionalita della versione commerciale

La versione di Corel Wordperfect 8 liberamente scaricabile da Internet **manca** di alcune parti presenti invece nella versione commerciale, e quindi a pagamento, del prodotto.
Si tratta sostanzialmente dell'editor di **grafici** e disegni da inserire poi nel documento o dell'editor di **formule** e **equazioni matematiche**, funzionalità quest'ultima particolarmente apprezzate in ambito accademico e scientifico.



Corel Wordperfect 8 per Linux

Negli ultimi anni la diffusione di Linux à aumentata in modo sensibile. E con l'aumentare degli utenti di questo sistema operativo, à cresciuto anche l'interesse intorno a esso da parte di nuove e vecchie software house.
Corel, azienda canadese famosa per la sua suite Corel Draw in ambiente Windows, sta scommettendo molto su Linux, giungendo anche a sviluppare, in accordo con Debian e il team di KDE, una distribuzione.
Parleremo qui di Corel Wordperfect 8, noto wordprocessor già esistente per altri sistemi operativi, di cui di recente à stata rilasciata una versione per Linux, la cui licenza à gratuita per usi non commerciali.

Dove trovarlo

E' possibile scaricare il file di installazione di Corel Wordperfect 8 dal sito http://linux.corel.com; il pacchetto ha dimensioni di circa 23 Mb.
E' comunque presente sul CD allegato, nella directory /office/wp, che useremo come punto di riferimento per la fa se di installazione. Sul CD-ROM sono presenti anche i moduli per l'italiano, sia per quanto riguarda il dizionario del controllo ortografico, sia per quanto riguarda il programma stesso: voci di menu, pulsanti, help in linea.

Installazione

Per procedere all'installazione à necessario aver montato il CD-ROM, così da consentirci di accedere al suo contenuto.
Procedere quindi come segue:

mount /dev/cdrom /mnt/cdrom

Dove /dev/cdrom à un link simbolico al device del lettore CD-ROM, e /mnt/cdrom à una directory che useremo come mountpoint, entro la quale ci troveremo il contenuto dal CD stesso. Il comando mount dovrà essere eseguito da utente root.

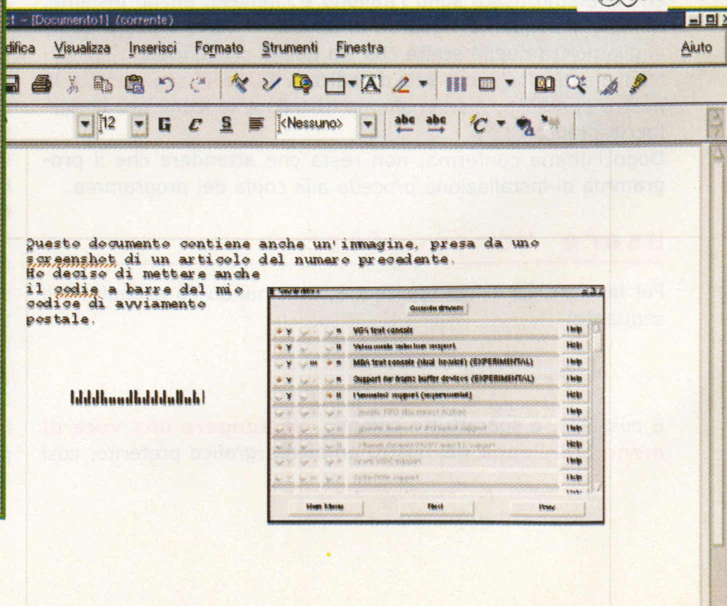
Difetti riscontrati

I difetti riscontrati nell'utilizzo di Corel Wordperfect 8 sono sostanzialmente di natura **tecnica**: è capitato durante le nostre prove, anche se molto di rado, che il programma si **crashasse** improvvisamente senza alcun motivo apparente.
Migliorabili i filtri di importazione da formati di documenti non Wordperfect: purtroppo vengono spesso persi o interpretati male i codici di controllo, ottenendo un documento completo nei suoi contenuti ma in buona parte senza la formattazione né gli attributi originali.
Inoltre vi sono casi apparentemente senza un comune denominatore in cui l'importazione del documento porta al **blocco** totale dell'editor, che può comunque essere risolto con l'invio del segnale SIGKILL (quindi non va ad inficiare la sicurezza del sistema). Un aspetto curioso che abbiamo notato è l'eccessivo consumo di CPU da parte del programma, anche in momenti di totale inutilizzo e senza alcun documento caricato. Le risorse di CPU tendono infatti ad essere utilizzate da Wordperfect per tutto il tempo disponibile, raggiungendo assurdi picchi del 99%.
Nonostante questo, le prestazioni della macchina **non** ne risentono negativamente, anche perché, forse, la memoria consumata si mantiene entro limiti ragionevoli.

Conclusioni

Malgrado i pochi difetti riscontrati, si può considerare Corel Wordperfect 8 un **buon wordprocessor** per Linux. Si configura come un candidato nell'aiuto per l'abbandono definitivo di altri sistemi operativi che siamo costretti a usare solo per motivi di **interscambio** di documenti: in questa direzione infatti sono orientati gli sforzi della casa produttrice, cercando una migliore e più totale compatibilità con i formati presenti e molto usati (Microsoft Word in testa).
Semplice da usare e decisamente intuitivo, si può imparare ad usare in pochi minuti, utilizzando una interfaccia ormai standard di fatto in questo tipo di prodotti. Il fatto che dietro ad un simile prodotto vi sia una azienda come Corel, è sicuramente una garanzia di assistenza ed upgrades.





Questo documento contiene anche un'immagine, presa da uno screenshot di un articolo del numero precedente.
Ho deciso di mettere anche
il codie a barre del mio
codice di avviamento
postale.





Collegarsi ad Internet

(manuale, con Gnome, con KDE)

Lorenzo Lazzeri
l.lazzeri@oltrelinux.com

Programmatore C/C++, si occupa di amministrazione di sistema persso un ISP. Sviluppa applicazioni Web con PHP e SQL server.

Connettere ad internet il nostro Linux appena installato è desiderio di molti. La cosa però presenta alcune difficoltà facilmente superabili: vediamo in dettaglio come effettuare la configurazione sia in modo manuale, che utilizzando tool grafici per Gnome e KDE.

Una volta installato il nostro Linux, la prima cosa che vorremmo fare è annunciare al mondo intero il risultato delle nostre fatiche. Purtroppo manca ancora un piccolo passo prima di poter procedere a questo annuncio: la configurazione per il collegamento a Internet. Molto spesso si sente dire che questa è un'impresa ardua rispetto a quanto lo è su altri sistemi operativi. Vedremo che non è affatto così e che molti tools volendo ci vengono in aiuto.

Requisiti

Per poter sfruttare il collegamento a Internet, è necessario che la nostra Linuxbox sia in grado di supportarne i **protocolli** utilizzati. Inoltre, serve un tool specifico, **pppd**, che si occuperà di stabilire la connessione.
Nel kernel, deve essere abilitato il supporto per i protocolli **TCP/IP** e **PPP**, più i vari metodi di compressione per quest'ultimo. La modifica di tali opzioni esula dallo scopo di questo articolo in quanto tale argomento sarà trattato nel dettaglio in futuro. Normalmente, comunque, il **kernel standard** delle varie distribuzioni è già impostato per il supporto di tali protocolli. Può capitare comunque che tali driver siano compilati come moduli da caricare a runtime tramite il comando **modprobe** (vedere la manpage per maggiori dettagli).

Chi vuol fare da solo

Il modo più veloce per consentire al proprio sistema di collegarsi a Internet consiste nel mettere le mani **direttamente** nei file di configurazione. Non è un metodo alla portata di tutti, soprattutto per via delle **numerose opzioni** spesso dai nomi quantomeno inusuali.
La directory /etc/ppp contiene diversi file necessari a pppd, nei quali sono contenuti dalle password per l'accesso, alle opzioni specifiche per la connessione.
Il primo file da modificare è /etc/ppp/options, in tale file dovranno essere inserite, **una per riga**, le opzioni di pppd che vogliamo vengano prese in considerazione ad ogni collegamento.
Un tipico elenco di tali opzioni, che si può tranquillamente copiare direttamente nel proprio file, contiene quanto segue:

```
lock
defaultroute
noipdefault
modem
/dev/modem
115200
crtscts
passive
asyncmap 0
name "rspider"
```

Vediamone nel dettaglio alcune: l'opzione **defaultroute** dice al pppd di impostare le tabelle di routing per i pacchetti in entrata e in uscita come verrà suggerito dal server a cui ci connettiamo. **/dev/modem** indica a quale device è collegato il modem. Tale device non è altro che un link simbolico al vero device, creato in questo modo:

```
# ln -s /dev/ttyS0 /dev/modem
```

Dove /dev/ttyS0 corrisponde alla COM1, /dev/ttyS1 a COM2, e così via.

Tra le altre opzioni notiamo **115200** che non è altro che la velocità della seriale. Importante è l'opzione denominata name "**rspider**", che dice al pppd con **quale nome** utente stiamo cercando di collegarci. E la password? La password per tale nome utente è contenuta in un'altro file, quello che andiamo ad analizzare, **/etc/ppp/pap-secrets** con il seguente formato:



Configurazione

```
"rspider"     *     "password"
```

Al posto di "**password**" ovviamente dovremo inserire la nostra. In questo file sono quindi indicate le password da inviare secondo il protocollo PAP (Password Authentication Protocol) per l'autenticazione dell'utente.

Un'altro file importante, quello che si occuperà di istruire pppd (anche se in realtà pppd stesso si appoggia ad un altro tool, chat) su come comunicare col modem, è **/etc/ppp/pppscript**:

```
TIMEOUT 60
ABORT ERROR
ABORT BUSY
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
"" "AT&FH0"
OK "atdt055700645"
TIMEOUT 75
CONNECT
```

In questo modo si indica a chat di impostare il **timeout** per le risposte dal modem a 60 secondi, di interrompere la connessione se si riceve dal modem stesso uno dei messaggi ERROR, BUSY, NO CARRIER o NO DIALTONE.
Tale file può anche essere **utilizzato** per collegarsi a quei (pochi) **provider** che necessitano un'autenticazione diversa da quella PAP. Su ogni riga, infatti, viene scritta la **stringa** che ci si **aspetta** dalla seriale (e quindi sia dal modem, sia dal server remoto una volta connessi) e, **separata da uno spazio**, quella da inviargli.
Aggiungendo infatti le seguenti righe:

```
login:          "rspider"
password:       "password"
```

Si indica al chat di rispondere col nome utente quando si riceve la richiesta di login, e di rispondere con la password alla richiesta della stessa.

L'ultimo file che è necessario modificare è quello che serve a impostare i server DNS: **/etc/resolv.conf**. Questo file non si trova nella solita directory /etc/ppp poiché le impostazioni ivi contenute non sono specifiche dell'utilizzo con il pppd.
In ogni caso, in /etc/resolv.conf si specifica il dominio di riferimento del nostro provider, cosa non necessaria, e i server DNS da utilizzare per risolvere i nomi dei siti:

```
search cosmos.it
nameserver 193.70.6.2
nameserver 193.70.6.3
```

Si possono inserire fino a **tre** nameserver, anche se nessun errore verrà comunicato nel caso di un **numero maggiore**, quelli oltre il terzo verranno silenziosamente **ignorati** e quindi **non utilizzati**.

Tutto è quindi pronto per collegarsi a Internet, manca solo di sapere il modo di dire a Linux di farlo!

Il comando da eseguire è il seguente:

```
/usr/sbin/pppd -detach connect
        "/usr/sbin/chat -v -f
                /etc/ppp/pppscript
```

Pppd, **automaticamente**, userà le opzioni contenute in /etc/ppp/options, il nome utente e la password contenuti in /etc/ppp/pap-secrets, e si appoggerà al programma chat per colloquiare col modem, utilizzando quanto inserito in /etc/ppp/pppscript per farlo.
Può essere conveniente **includere** quel **comando** in uno script di shell dal nome più **corto**, così che non si debba ricordare tutte le volte quella serie di parametri.
Bisogna però tenere presente che il file /usr/sbin/pppd può normalmente essere eseguito solo dall'utente **root**, se si vuole consentire l'accesso a Internet anche a un utente normale (caso comune per PC casalinghi), si devono modificare i permessi con il comando **chmod**:

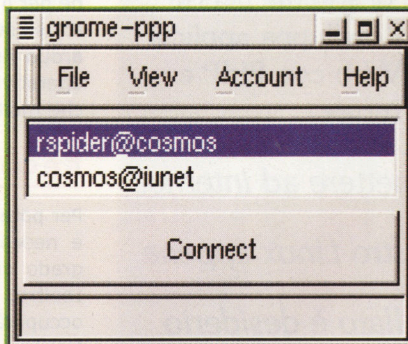
```
chmod 755 /usr/sbin/pppd
```

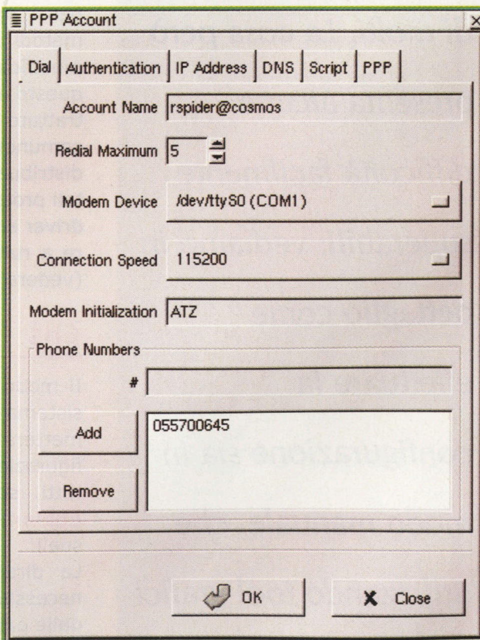
Per chi vuole essere aiutato

Esistono metodi **alternativi** per collegarsi a Internet, soprattutto con l'avvento degli ambienti KDE e GNOME, che vengono installati con dei tool appositi. Tool che **imitano** il più famoso "Accesso Remoto" di Windows 95, molto spesso estendendone le funzionalità e addirittura la semplicità d'uso.
L'unico neo dell'utilizzare questi sistemi consiste nel fatto che bisogna avere l'ambiente utilizzato in **esecuzione**, ma per un sistema desktop su cui molto spesso GNOME o KDE costituiscono l'interfaccia di default, questa non è una **grossa limitazione**.
I tool che prenderemo in considerazione sono gppp e kppp, il primo fa parte di GNOME, il secondo di KDE.

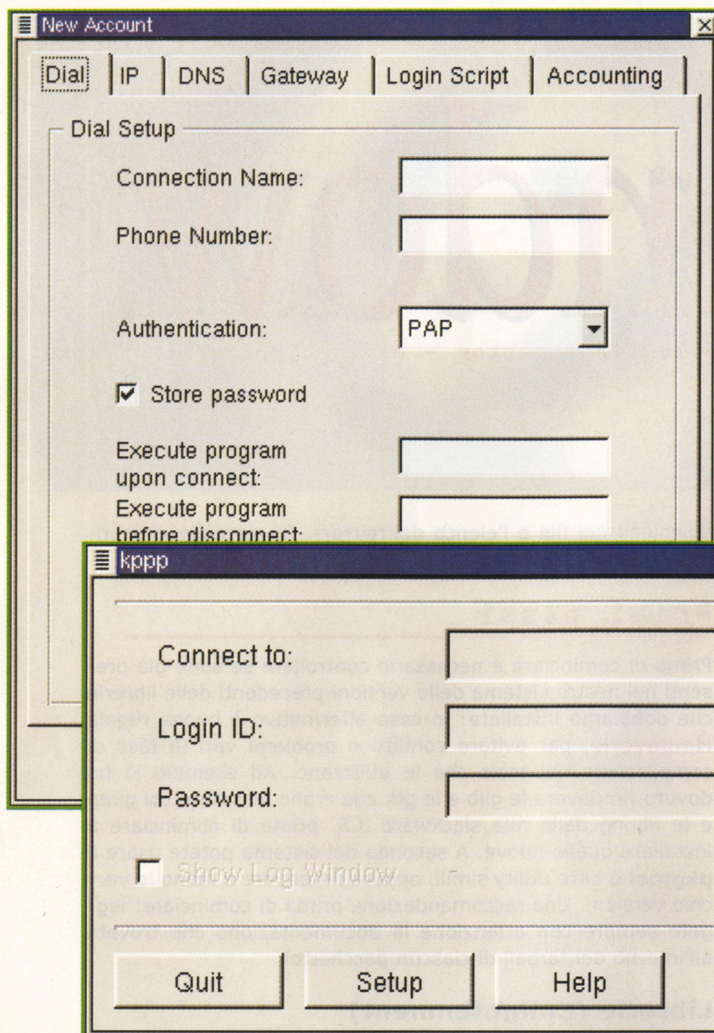
GPPP

Gppp viene lanciato scegliendo l'opzione "PPP dialup utility" dal menu "Internet" del main menu di GNOME.
Una volta aperto si dovrà configurare la prima connessione, scegliendo dal menu "Account" dell'applicazione la voce "New...".









Le varie fasi della configurazione usando KDE: le immagini sono davvero autoesplicative.

La finestra che si apre è suddivisa per argomenti in schede, in ognuna delle quali si dovranno inserire i parametri richiesti per il collegamento. Nella scheda "Dial" verrà chiesto di dare un **nome mnemonico** alla connessione che si va creando, nonché il numero di telefono del provider (eventualmente più di uno), la stringa di inizializzazione del modem, la seriale a cui questo è collegato. Nella finestra "Authentication" invece sono richiesti il nome utente (nel campo "User Name") e la password (nel campo "Hidden Password"). Il campo "Remote Name", tranne in casi particolari in cui il metodo di autenticazione sia il CHAP, dovrà essere lasciato in bianco. Le opzioni di default presentate nella scheda "IP Address", nella maggior parte dei casi sono adatte. Esse indicano a gppp che sia il nostro indirizzo IP, sia la netmask, sia l'indirizzo IP del server remoto, verranno assegnati dinamicamente in fase di connessione.

La scheda "DNS" invece consente di specificare il dominio del provider e i name server da usare per la risoluzione dei nomi, mentre nella scheda "Script" è possibile creare uno script di connessione nel caso di autenticazione tramite richiesta di login e password in modalità terminale.

L'ultima scheda, "PPP", presenta opzioni di default adatte alla maggior parte dei casi, nonché la possibilità di specificare un programma da eseguire dopo la connessione e uno da eseguire dopo la disconnessione.

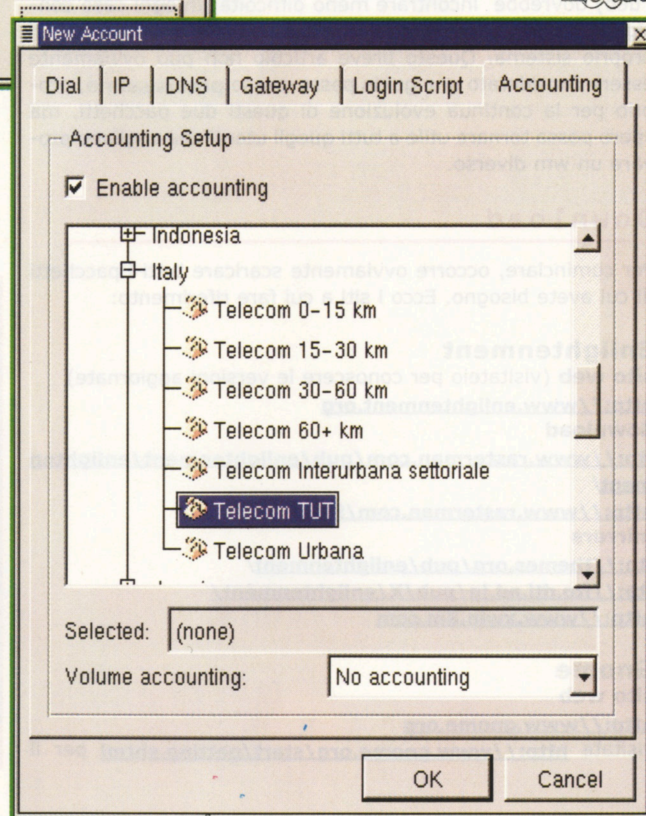
Per effettuare il collegamento, una volta eseguito gppp, si dovrà premere sul pulsante "Connect".

KPPP

Kppp, il tool per la connessione a Internet dell'ambiente KDE, è molto simile a Gppp per quanto riguarda le opzioni richieste.

Una volta eseguito, premendo sul pulsante "Setup" si accede alla lista degli account creati. Si possono modificare le opzioni per un account esistente premendo il pulsante "Edit...", oppure crearne uno nuovo clickando sul bottone "New...". In questo caso si apre una finestra simile a quella di Gppp, divisa in schede a seconda del gruppo di opzioni. E' interessante la possibilità offerta dalla scheda "Accounting": una volta indicato il paese da cui si effettua la connessione e la tariffa telefonica di appartenenza, Kppp si occupa di indicarci costantemente quanto stiamo spendendo, così da tenerci informati sui costi. Una volta finita la configurazione si puo' utilizzare Kppp per collegarsi clickando sul pulsante "Connect".





X-window

Marco Ivaldi - raptor - m.ivaldi@oltrelinux.com

Enlightenment e Gnome sono a mio avviso i due ambienti X Window più stimolanti e piacevoli da guardare e da utilizzare. Una loro pecca, però, è che ora come ora non è facile reperire documenti che spieghino come **installarli effettivamente**, trattino dei passi che è necessario fare e insomma suggeriscano qualche piccolo trucchetto che a volte può far evitare la perdita di **ore** di tentativi davanti al computer. Gnome ed E sono ancora decisamente in via di sviluppo e vengono costantemente aggiornati e modificati, ma personalmente posso dire che Enlightenment non è poi tanto meno stabile di KDE, per fare un esempio. Quello che segue è il risultato degli appunti presi durante l'installazione di E 0.15.5 sulla mia slackware 3.5 e della consultazione di docs e FAQ sul web, insieme a una buona dose di tentativi. Tratterò dell'installazione a partire dai sorgenti; se qualcuno avesse la possibilità di usare gli **rpm** (o i deb) dovrebbe incontrare meno difficoltà: in ogni caso consiglio sempre di scaricare i tar.gz e compilare i sorgenti sul proprio sistema. Questo breve articolo non può ovviamente essere considerato una guida passo-passo **precisissima**, proprio per la continua evoluzione di questi due pacchetti, ma spero possa tornare utile a tutti quegli utenti che vogliono provare un wm diverso.

Download

Per cominciare, occorre ovviamente scaricare tutti i pacchetti di cui avete bisogno. Ecco i siti a cui fare riferimento:

Enlightenment

sito web (visitatelo per conoscere le versioni aggiornate)
http://www.enlightenment.org
download
ftp://www.rasterman.com/pub/enlightenment/enlightenment/
http://www.rasterman.com/files/
mirrors
ftp://themes.org/pub/enlightenment/
ftp://ftp.dti.ad.jp/pub/X/enlightenment/
http://www.xwin.8m.com

Gnome

sito web
http://www.gnome.org
visitate http://www.gnome.org/start/getting.shtml per il download dei file e l'elenco dei mirrors. Io consiglio di scaricare i tarball, piuttosto che i file precompilati.

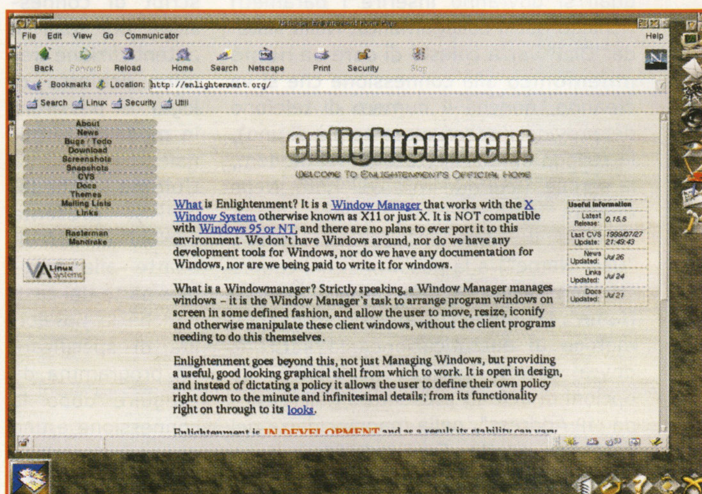
Primi passi

Prima di cominciare è necessario controllare se sono già presenti nel nostro sistema delle versioni precedenti delle librerie che dobbiamo installare: in caso affermativo è buona regola **rimuoverle**, per evitare conflitti e problemi vari in fase di compilazione dei tools che le utilizzano. Ad esempio io ho dovuto rimuovere le glib e le gtk che erano installate col gimp e le libpng dalla mia slackware 3.5, prima di cominciare a installare quelle nuove. A seconda del sistema potete usare il **pkgtool** o altre utility simili, oppure rimuovere a mano le vecchie versioni. Una raccomandazione prima di cominciare: leggete sempre con attenzione la documentazione che trovate all'interno dei tarball di ciascun pacchetto!

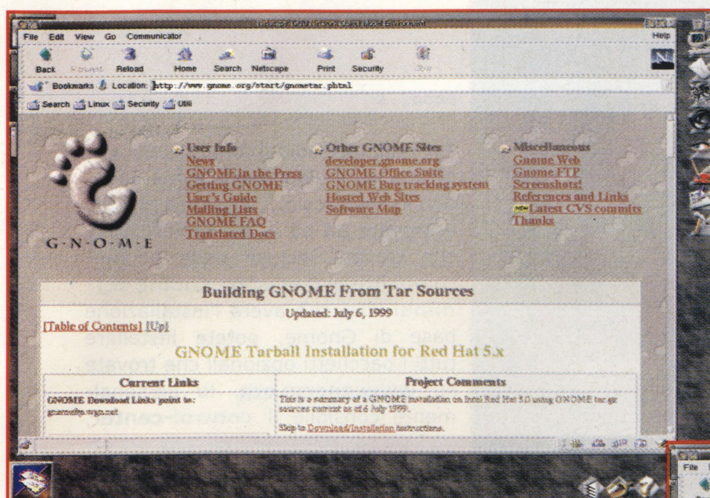
Librerie (Enlightenment)

Questo è schematicamente l'ordine in cui installare i pacchetti che vi servono:

zlib - libpng - libgif o libungif - libjpeg - freetype - glib - gtk - imlib - fnlib - esound (per chi vuole il supporto per il suono)







Generalmente ciò si riduce ad un semplice:

```
./configure ; make; make install
```

In alcuni pacchetti però possono esserci delle altre opzioni a disposizione e a questo proposito vi consiglio di leggere in modo accurato i file README e INSTALL che trovate all'interno di ogni tar.gz.



Alcune osservazioni.

La cosa forse più difficoltosa da installare in questa fase sono le **glib** e le **gtk**. Innanzitutto esse devono essere della **stessa versione**, in quanto originariemente facevano parte di uno stesso pacchetto che poi è stato diviso nel corso dello sviluppo del software. In ogni caso dopo esservi assicurati che nel vostro sistema non ci siano versioni più vecchie già installate, procedete alla compilazione e all'installazione delle glib e successivamente delle gtk. Può risultare necessario aggiungere il nuovo path in /etc/profile, in questo modo:

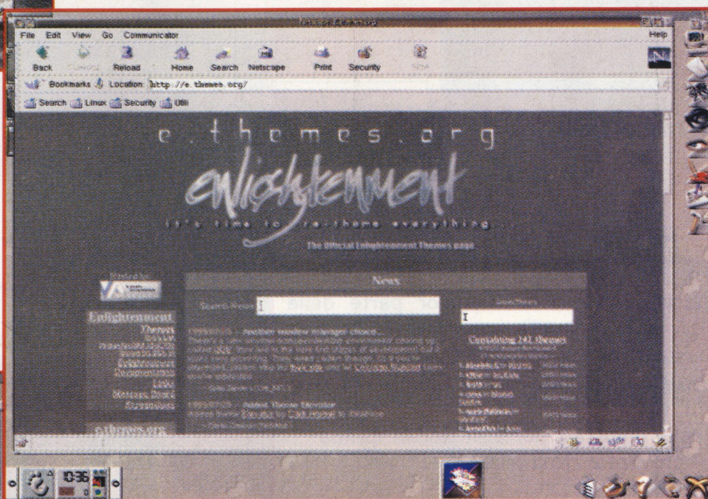
```
# gtk path
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Per quanto riguarda le imlib, esse hanno bisogno delle altre librerie che vengono installate prima, per funzionare correttamente. Nell'ordine: **zlib**, **libpng**, **libgif** o **libungif** (*a scelta una delle due*), **libjpeg**, **freetype**, **glib**, **gtk**.

Se infine volete abilitare il supporto per il **suono** installate anche il pacchetto **esound**.

Enlightenment

A questo punto non vi resta che compilare l'enlightenment vera e propria: una volta installata con make install, se tutto



è andato a buon fine potete aggiungere alla fine del file .xinitrc che si trova nella home dell'utente che userà X la riga:

```
exec /usr/local/enlightenment/bin/enlightenment
```

assicurandovi che le righe precedenti non contengano il comando exec. Se .xinitrc non esiste, potete crearlo rapidamente con il seguente comando:

```
echo "exec /usr/local/enlightenment/bin/enlightenment"
    > $HOME/.xinitrc
```

Ovviamente tutto ciò assumendo che abbiate installato E nella sua directory of default (/usr/local/enlightenment). Ora basta avviare normalmente X window con un semplice startx. Congratulazioni! Ecco a voi Enlightenment.

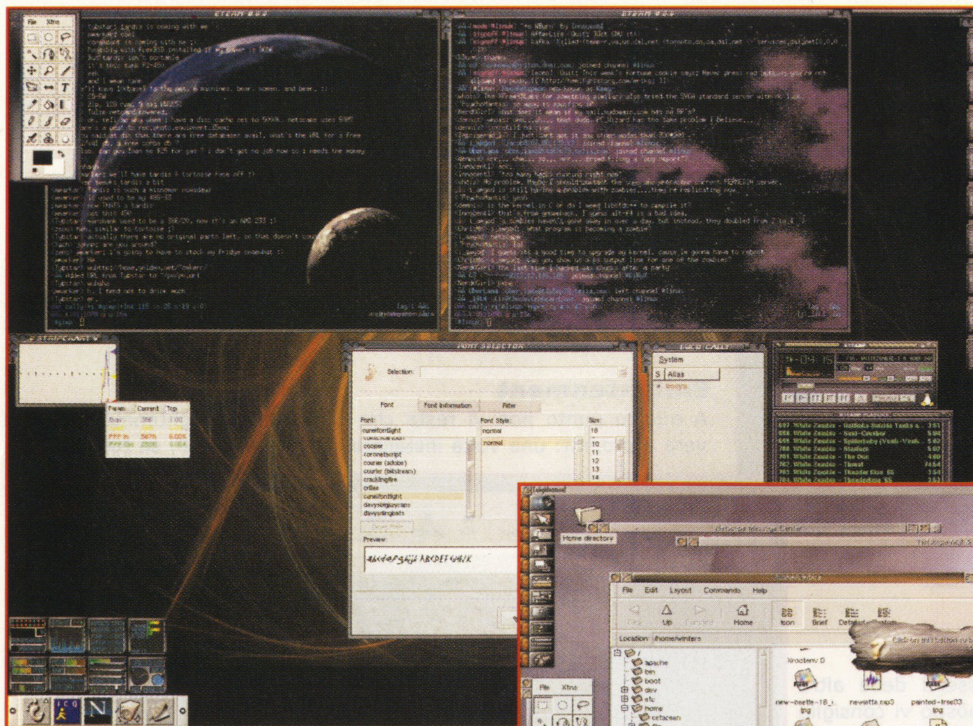
Librerie (Gnome)

Come ho fatto per enlightenment, ecco qui di seguito riportati i pacchetti che vi servono e l'ordine in cui installarli. Alcuni sono stati omessi, in quanto già compresi nell'installazione di E:

audiofile - ORBit - gsl - gtk-engines - guile - libGtop - gnome-libs



Configurazione






```
panel &
```

Per utilizzare solo il famoso pannello di Gnome. Quindi lanciate X e godetevi lo spettacolo... leggetevi gli help e provate un po' delle applet carine che trovate incluse nelle gnome-core, oltre al fantastico **gnome-terminal**. Ora che avete l'installazione base di Gnome, potete installare tutti i pacchetti opzionali che trovate su www.gnome.org. Io personalmente consiglio il **control-center**,

Anche qua per la maggior parte delle librerie che dovete compilare e installare basta un semplice `./configure ; make ; make install` . Se doveste avere problemi, molte di queste librerie permettono il make uninstall, per disinstallare prima di provare a ricompilare.

Osservazioni

La cosa principale a questo punto e' l'installazione delle librerie che richiedono le gnu **gettext** (come ad esempio le gnome-libs, cfr ai file README e INSTALL dei singoli pacchetti per sapere quali altre ne hanno bisogno). In realtà è possibile compilare tutto agevolmente senza installare le gettext, ed è questo il metodo che vi consiglio. Basterà semplicemente lanciare lo script di configure in questo modo:

```
./configure --disable-nls --with-gnu-gettext
```

Così il make linkerà le librerie comprese nei singoli pacchetti senza andare a cercare l'installazione delle gettext. Molto comodo e da tenere presente, perchè spesso le librerie shared delle gettext danno dei problemi in questa fase.

Gnome

Ci siamo quasi, ancora un piccolo sforzo... Compilate e installate gnome-core, e aggiungete la riga seguente nel file .xinitrc, PRIMA di quella aggiunta precedentemente per E:

```
gnome-session &
```

Oppure semplicemente:

che serve come supporto all'installazione di **e-conf**, il programma molto comodo per configurare molti aspetti di enlightenment.

Temi

Troverete tutti i temi per Enlightenment disponibili su http://e.themes.org , non perdeteveli! Alcuni screenshots sono davvero spettacolari.

Documentazione

Fate riferimento ai due siti
http://enlightenment.org
http://www.gnome.org
per tutta la documentazione e le applicazioni di cui avete bisogno. **Buon divertimento!**



FreeBSD



Emanuele Busuito
e.busuito@oltrelinux.com

Studente di Ingegneria Informatica, è un appassionato di sicurezza informatica e programmazione in ambito freeUNIX (Linux, *BSD)

FreeBSD è tra i sistemi operativi più stabili e performanti oggi disponibili. Una breve introduzione sulle versioni disponibili e una guida all'installazione e alla configurazione, che proseguirà per alcune puntate.

Se andassimo a chiedere cos'è FreeBSD in un canale IRC su Linux o su un newsgroup probabilmente ci risponderebbero che è un sistema UNIX-Like poco conosciuto e ne sparlerebbero mettendo in luce le qualità di Linux a fronte dei difetti di FreeBSD (ma ne esistono?).

Questo perchè tra la comunità "Linuxiana" e "FreeBSDiana" c'è sempre stata una diatriba su quale fosse il migliore sistema operativo tra i due, ma spesso queste discussioni si trasformano in inutili flames (litigi) che non portano a nessun risultato.

È difficile determinare quale sia il migliore tra i due e alla fine le differenze sono più filosofiche che tecniche, perchè entrambi sono ottimi sistemi, veloci stabili ed efficenti e con un corredo di free software eccezionale.

Cos'è FreeBSD ? Semplice! È un sistema operativo derivato da **BSD** UNIX, **completamente gratuito** e con uno sviluppo **centralizzato**, a differenza di Linux che viene diviso in diverse distribuzioni.

Per farci una idea della sua potenza, vi faccio notare che **YAHOO** (uno tra i piu' famosi motori di ricerca) lo utilizza nei suoi numerosi server e **Walnut Creek CDROM** (il distributore di Linux Slackware e FreeBSD) lo utilizza come server ftp su una macchina ultrapotente con 1 GB di ram e 150 GB di hard disk. Se FreeBSD viene usato su server di questo calibro un motivo ci dovrà essere...

Esistono **tre versioni** di FreeBSD (*di BSD si parla solo di versioni!*), che vengono utilizzate da diversi tipi di utenza in base ai loro scopi, eccole qui di seguito elencate:

FreeBSD-RELEASE : è l'ultima versione rilasciata per un uso generale, che contiene alcune nuove features ma è **stabile** e sottoposta a molti test prima di essere distribuita; questa è la versione maggiormente utilizzata ed è quella che di solito viene venduta in molti siti internet.
Versione corrente: 3.3

FreeBSD-STABLE : è l'ultima versione di FreeBSD-RELEASE a cui sono stati applicate tutte le possibili **correzioni** per renderla il più stabile possibile, non è detto che sia basata sull'ultima versione di FreeBSD-RELEASE, anzi, di solito è basata su **versioni** più vecchie che vengono migliorate.
Versione corrente: 2.2.8

FreeBSD-CURRENT : è l'ultima versione di FreeBSD che è ancora in via di sviluppo ed è molto **instabile** perchè contiene nuovi cambiamenti sperimentali che verranno poi introdotti nelle successive versioni CURRENT.
Versione corrente: 4.0

Installiamo...

L'installazione di FreeBSD è semplice ed è simile a quella delle distribuzioni Linux (in particolare a quella della Slackware) e quindi, chiunque abbia già installato Linux si troverà a suo agio: per gli utenti provenienti dal mondo Windows basterà eseguire attentamente le operazioni di seguito elencate.

L'installazione può essere effettuata tramite CD-ROM, Server ftp, Server NFS, e anche tramite una partizione DOS, FreeBSD e persino da floppy (*sconsigliato in ogni caso*) o tape.

Se il vostro PC supporta il boot da CDROM e possedete la copia originale di FreeBSD o comunque il cd in vostro possesso è bootabile potete far partire l'installazione direttamente da CDROM, altrimenti vi occorrono due floppy disk (uno di boot e uno per il root fs), le immagini sono contenute nella sottodirectory floppies del cdrom, i file sono **kern.flp** e **mfsroot.flp**.

Per trasferire questi dati su floppy da DOS o Windows basterà utilizzare il programma **fdimage** contenuto nella sottodirectory tools. Basterà posizionarsi nella directory tools del



Spazio BSD

```
---------------------- Choose Distributions ----------------------
As a convenience, we provide several "canned" distribution sets.
These select what we consider to be the most reasonable defaults for the
type of system in question.  If you would prefer to pick and choose the
list of distributions yourself, simply select "Custom".  You can also
pick a canned distribution set and then fine-tune it with the Custom item.

Choose an item by pressing [SPACE]. When you are finished, chose the Exit
item or press [ENTER].

 [ ] 1 Developer          Full sources, binaries and doc but no games
 [ ] 2 X-Developer        Same as above + X Window System
 [ ] 3 Kern-Developer     Full binaries and doc, kernel sources only
 [ ] 4 X-Kern-Developer   Same as above + X Window System
 [ ] 5 User               Average user - binaries and doc only
 [ ] 6 X-User             Same as above + X Window System
 [ ] 7 Minimal            The smallest configuration possible
 > > 8 Custom             Specify your own distribution set
 v(+)

                 [  OK  ]    Cancel
------[ Press F1 for more information on these options. ]------
```

Se ci sentiamo del hackers selezioniamo X-Kernel-developer ;)

cdrom e dal PROMPT di ms-dos digitare:

```
D\tools>
 fdimage ..\floppies\kern.flp a:
```

e successivamente

```
D\tools>
 fdimage ..\floppies\mfsroot.flp a:
```

(assumendo che **D** sia la lettera del lettore CD e **A** quella del floppy).
Mentre se ci troviamo in un sistema UNIX (e quindi anche in Linux) basterà **montare** il cdrom, posizionarci nella directory /cdrom/FreeBSD e digitare:

```
PunkZone:/cdrom/floppies#
  dd if=kern.flp of=/dev/fd0
```

e successivamente:

```
PunkZone:/cdrom/floppies#
  dd if=mfsroot.flp of=/dev/fd0
```

ATTENZIONE: cambiate i dati con quelli della vostra macchina, io ho **"montato"** il cdrom nella directory /cdrom (mount -tiso9660 /dev/cdrom /cdrom) e utilizzo come periferica floppy /dev/fd0.

Se la copia è avvenuta correttamente dovrebbe apparire un messaggio simile:

```
2880+0 records in
2880+0 records out
```

Mentre in caso di errori apparirà il messaggio:

```
dd: /dev/fd0: I/O error
```

Adesso siamo pronti per iniziare l'installazione, se non avete nessuna partizione libera vi conviene utilizzare programmi come Partition Magic o come FIPS che è contenuto nella directory tools; ricordatevi prima di utilizzarlo di leggere attentamente il file tools\fips.doc.
Questi programmi **riducono** la dimensione delle partizioni VFAT e DOS utilizzando lo **spazio libero** per creare **nuove** partizioni e poter così usare file system e sistemi operativi differenti.
Attenzione effettuate prima un backup dei dati contenuti nella partizione perchè questa operazione potrebbe **distruggere** tutti i dati in essa contenuti.
Adesso resettiamo il computer e inseriamo il floppy contenente l'immagine boot.flp nel drive (ovviamente deve essere impostato nella sequenza di boot del BIOS il drive del floppy).
A questo punto dovrebbe partire automaticamente il disco di boot, dopo alcuni secondi apparirà il messaggio:

```
Please insert MFS root floppy and
press enter:
```

A questo punto togliamo il boot disk e inseriamo il floppy contenente l'immagine mfsroot.flp e premiamo INVIO, dopo alcuni secondi di caricamento dovrebbe apparire un menu:

Kernel Configuration Menu

- Skip kernel configuration and continue with installation
- Start kernel configuration in full-screen visual mode
- Start kernel configuration in CLI mode

in questa fase si può configurare il kernel di boot di FreeBSD in modo da supportare tutto l'hardware che possediamo, nella maggior parte dei casi basterà premere INVIO, mentre se vogliamo configurare alcune periferiche spostiamo coi cursori sulla seconda opzione e premiamo INVIO.
Adesso ci appariranno in sequenza le periferiche che vengono riconosciute dal Kernel e successivamente partirà auto-

```
Disk name:     wd0                                  FDISK Partition Editor
DISK Geometry:  525 cyls/255 heads/63 sectors = 8434125 sectors

   Offset       Size        End     Name   PType      Desc  Subtype    Flags

        0         63         62        -       6    unused        0
       63    4080447    4080509   wd0s1        3   freebsd      165       C
  4080510    4358340    8438849        -       6    unused        0

The following commands are supported (in upper or lower case):

A = Use Entire Disk     B = Bad Block Scan       C = Create Slice
D = Delete Slice        G = Set Drive Geometry   S = Set Bootable
T = Change Type         U = Undo All Changes     W = Write Changes

Use F1 or ? to get more help, arrow keys to select.
```

Ecco come appare fdisk, il programma per ripartizionare i dischi

```
FreeBSD Disklabel Editor

Disk: wd0     Partition name: wd0s1    Free: 831 blocks (0MB)

Part      Mount          Size Newfs   Part      Mount          Size Newfs
----      -----          ---- -----   ----      -----          ---- -----
wd0s1b    swap          100MB SWAP
wd0s1a    <none>       1892MB *

The following commands are valid here (upper or lower case):
C = Create      D = Delete        M = Mount pt.  W = Write
N = Newfs Opts  T = Newfs Toggle  U = Undo       Q = Finish
A = Auto Defaults for all!

Use F1 or ? to get more help, arrow keys to select.
```

Non ci rimane che decidere come dividere la nostra partizione FreeBSD

maticamente il programma di installazione con una semplice interfaccia grafica in modalità testo.
Per prima cosa se vogliamo utilizzare la tastiera italiana, spostiamoci coi cursori sull'opzione **"K - Keymap"** e scegliamo la tastiera italiana (*italian ISO keymap*) premendo SPAZIO sull'opzione apposita e poi INVIO.
Per installare FreeBSD possiamo scegliere 3 opzioni in base al nostro grado di conoscenza del sistema:

- **Novice installation**: per chi non conosce FreeBSD.
- **Express installation**: per chi vuole installarlo velocemente e facilmente
- **Custom installation**: per utenti esperti che hanno una certa padronanza del sistema.

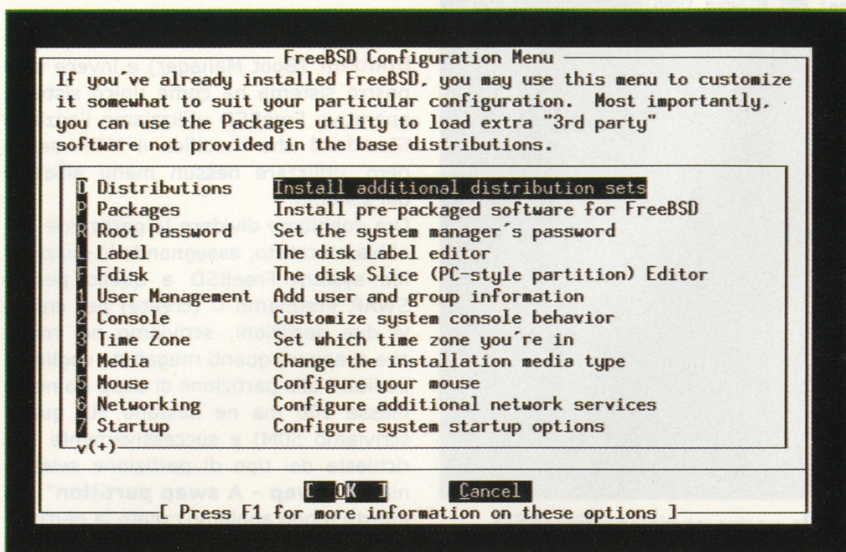
Noi utilizzeremo l'installazione **Express** per semplificare e velocizzare il nostro lavoro, quindi posizioniamoci sull'opzione Express e premiamo INVIO.
A questo punto ci apparirà l'editor dei dischi (fdisk), in cui noi dobbiamo decidere quale partizione utilizzare per il file system FreeBSD e lo swap (*ovvero la partizione per la memoria virtuale*).
Se abbiamo una partizione libera già pronta ci basterà utilizzare l'opzione **C** (*Create slice*) specificando in Megabyte la dimensione che vogliamo utilizzare (per esempio per specificare un Gigabyte scriviamo 1000M). Alla richiesta del tipo di partizione da creare premiamo semplicemente INVIO in modo da creare un partizione di tipo 165 (ovvero FreeBSD native fs). Mentre se vogliamo cancellare qualche partizione che non ci serve più spostiamoci sulla partizione in questione con i cursori e premiamo D (Delete slice).
Finite le nostre operazioni sui dischi premiamo Q per uscire.
Adesso ci verra richiesto come dovrà partire FreeBSD al prossimo riavvio, se possediamo altri sistemi con boot manager integrato (Linux, NT...) utilizziamo l'opzione **None** (ricordiamoci poi di modificare il boot manager degli altri sistemi), mentre se vogliamo creare noi un menu di scelta all'avvio del sistema (per esempio se possediamo Windows 98) posizioniamoci sull'opzione **BootMgr** (Boot Manager) e invece se il nostro sistema ha come unico sistema operativo FreeBSD utilizziamo l'opzione **Standard** che modifica il **MBR** senza pero' utilizzare nessun menu all'avvio del sistema.
Ora dobbiamo dividere la partizione che abbiamo creato, assegnando lo spazio al file system FreeBSD a quello per lo SWAP. Premiamo **C** (create) per creare le due partizioni, scriviamo nel menu che ci appare quanti megabyte vogliamo dedicare alla partizione di swap (io ne ho messe 100 ma ne bastano 50, quindi scriviamo 50M) e successivamente alla richiesta del tipo di partizione selezioniamo "**Swap - A swap partition**" , in questo modo abbiamo creato la partizione di swap. Ora creiamo il root file system ancora con l'opzione C e assegnando come dimensione quella che viene riportata nella finestra di dialogo: XXXXX blocks (yyyyMB) are free, e questa volta selezioniamo "**FS - A file system**", e alla richiesta del mount point da utilizzare digitiamo /.
ATTENZIONE: Se FreeBSD deve essere utilizzato in un ambito aziendale sarebbe opportuno creare diverse partizioni per le directory /usr, /var ...
Quando abbiamo finito la ripartizione del disco premiamo Q.
Ora dobbiamo scegliere che tipo di sistema intendiamo utilizzare, per un uso casalingo è consigliato scegliere l'opzione **X-User** che contiene la maggior parte dei programmi di utilità di sistema

```
Choose Installation Media
FreeBSD can be installed from a variety of different installation
media, ranging from floppies to an Internet FTP server.  If you're
installing FreeBSD from a supported CDROM drive then this is generally
the best media to use if you have no overriding reason for using other
media.

1 CDROM        Install from a FreeBSD CDROM
2 FTP          Install from an FTP server
3 FTP Passive  Install from an FTP server through a firewall
4 DOS          Install from a DOS partition
5 NFS          Install over NFS
6 File System  Install from an existing filesystem
7 Floppy       Install from a floppy disk set
8 Tape         Install from SCSI or QIC tape
9 Options      Go to the Options screen

          [ OK ]          Cancel
[ Press F1 for more information on the various media types ]
```

Se vogliamo provare una esperienza "irripetibile" possiamo sempre decidere di effetturare l'installazione da floppy



Ecco come appare il menu di configurazione del sistema

compreso X-Windows, nulla ci vieta scegliere singolarmente ogni pacchetto utilizzando l'opzione Custom.
Scegliamo da quale periferica installare i pacchetti del sistema: noi utilizzeremo l'opzione CDROM.
Ai messaggi seguenti premiamo sempre INVIO: appariranno alcuni **avvisi** relativi al fatto che non abbiamo installato su **partizioni** diverse le **directory** più importanti.
In base alla velocità del nostro lettore CD e hard disk, dovremo aspettare alcuni minuti per la copia dei pacchetti nella partizione di destinazione.
Abbiamo installato FreeBSD. Prima di concludere alcuni suggerimenti su come configurare essenzialmente il sistema usando il programma sysinstall.

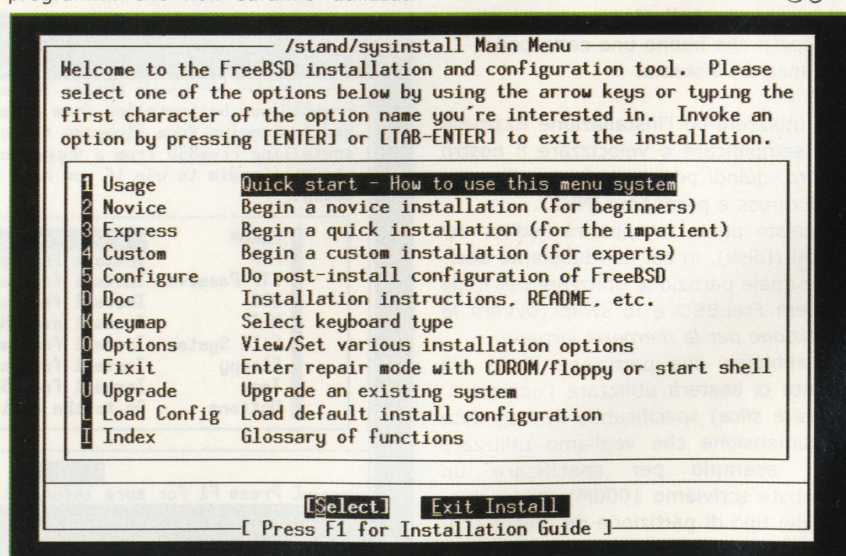
Sysinstall: un amico al tuo fianco!

Finita l'installazione del sistema ci verrà chiesto se vogliamo effettuare una **configurazione** "**post-installation**" del sistema: la risposta è ovviamenti si.
Prima di tutto selezioniamo R e settiamo la **password** che dovra' avere **root** ovvero il super-utente (amministratore) del nostro nuovo sistema.
Sarebbe utile anche aggiungere un utente **non privilegiato** selezionando 1, questo utente potrà poi essere utilizzato da noi per la **navigazione in internet**.
RICORDATE, non utilizzate l'utente root quando siete connessi ad internet perché, nel caso il programma che state usando presenti **bugs**, potreste ritrovarvi qualche ospite indesiderato in casa...
Altra cosa importante che dovete fare è decidere quali servizi devo essere avviati automaticamente ad ogni boot, per fare questo utilizziamo l'**opzione 7** (*System Startup Options*).
Avviate solo ciò che ritenete realmente necessario: è inutile sprecare risorse per programmi che non saranno utilizzati (ad esempio se non desiderate creare un server NFS -un protocollo di trasferimento dati- commentate la linea "This host wishes to be an NFS server").
Queste sono le **opzioni** più importanti che dovete settare, ma vi conviene dare un'occhiata a tutte le altre per raggiungere una conoscenza completa almeno delle principali funzioni.
Se desiderate configurare il vostro sistema anche dopo l'installazione basterà **loggarsi** come root e digitare /**stand/sysinstall**: vi riapparirà così il menu che avete utilizzato durante l'installazione.
Potete trovare qualsiasi informazione su FreeBSD al sito http://www.FreeBSD.ORG, mentre se volete scaricarvi interamente da internet l'ultima versione di FreeBSD dirigetevi al sito ftp.FreeBSD.ORG.
Ben ora avete un sistema UNIX tra le mani con le funzionalità tipiche dei **grandi** server internet: è importante che acquistiate padronanza del sistema in modo da potervi muovere in modo adeguato anche davanti a problemi.
Ricordatevi il primo comando che dovete digitare la prima volta che fate partire il sistema è

```
PunkZone:~#    man man
```

Così capirete la vera filosofia che sta dietro ai sistemi UNIX.





Bene siamo pronti ad iniziare l'installazione



INSTALLAZIONE DI UN FIREWALL

SALVATORE SANFILIPPO
s.sanfilippo@oltrelinux.com

Salvatore Sanfilippo è esperto di sicurezza delle reti, si occupa di programmazione distribuita, sviluppo di software per il controllo e la sicurezza.

*Il firewall è uno strumento molto utilizzato nei "programmi" di sicurezza aziendale. In una serie di articoli vedremo di capire cosa è un firewall, quale è il suo funzionamento, il livello di protezione che offre, cosa riesce a proteggere. Oltre al firewall saranno presentate le caratteristiche del protocollo Tcp/Ip: non è infatti possibile utilizzare al meglio un firewall senza sapere **cosa** si va a filtrare.*

Il firewall è, quasi certamente, lo strumento di **sicurezza informatica** che gode di **fama maggiore** tra le masse. Un altro primato del firewall è quello di essere, in assoluto, uno degli strumenti che danno una forte **"sensazione di sicurezza"**, spesso non giustificata. Per la logica comune, se un host ha problemi di sicurezza, installando un firewall si risolve tutto: **nulla di più falso**. Esistono più tipi di firewall, ma tutti, a prescindere dal tipo, dalla marca e dalla configurazione non sono altro che dei filtri. Spesso i firewalls sono classificati in due tipi: i proxy ed i packet filtering. Quello che differenzia queste due tipologie di firewall è **il livello a cui filtrano:** i firewalls **proxy** filtrano a livello della applicazione mentre i **packet filtering** filtrano a livello di trasporto e di rete (per una trattazione sulla stratificazione del protocollo TCP/IP vedi "*Internetworking with TCP/IP*" di Douglas E. Comer, Prentice Hall International). I packet filtering sono in assoluto i più **diffusi**, poichè anche se forniscono una protezione **minore**, sono estremamente più **flessibili**. L'altra motivazione che ha decretato il poco successo dei firewall proxy è il fatto che la tendenza moderna della sicurezza informatica è quella di affidare il compito di "filtrare" il livello applicativo all'applicazione stessa.

Per tutte queste ragioni questo articolo tratta dei packet filtering. Per arricchire la teoria con qualche applicazione saranno presentati degli esempi utilizzando "**ipchains**" e dunque il firewalling del **kernel 2.2.x** del sistema operativo Linux.

Il concetto di regola

Il firewall è un **filtro** e quindi, per poterlo utilizzare bene, è necessario conoscere perfettamente quello che si intende filtrare. Questa scelta è **fondamentale** e determina l'efficacia del firewall. Nel caso dei firewall proxy ad essere filtrati sono i pacchetti in entrata ed quelli in uscita. Un buon firewall ci dà la possibilità di decidere la sorte dei pacchetti in base alle seguenti **caratteristiche**:

Protocollo IP:

- Indirizzo sorgente
- Indirizzo di destinazione
- opzioni IP (non sempre purtroppo)
- protocollo
- frammentazione (bit more fragment settato?)

Protocollo ICMP:

- tipo ICMP
- codice ICMP

Protocollo UDP/TCP:

- porta sorgente
- porta destinazione
- flags: SYN, ACK eccetera (solo TCP)

Alcuni firewalls, come ad esempio ipfilter (*probabilmente il migliore in assoluto*), permettono di filtrare i pacchetti in relazione a tutti i campi dei protocolli supportati, anche in base allo "**stato**" delle connessioni. Anche se ipchains non raggiunge questo livello di dettaglio, costituisce un ottimo passo in avanti nei confronti del vecchio **ipfwadm** (in effetti ipchains ed ipfwadm sono i nomi dei programmi utilizzati per settare le regole. Parlando di "ipchains" si fa riferimento al firewalling del kernel 2.2, mentre "ipfwadm" indica il firewalling del kernel 2.0). La configurazione di un firewall non è altro che la **definizione delle "regole"** che determineranno la sorte del pacchetto. Eccovi un esempio di regola:

blocca tutti i pacchetti TCP uscenti destinati alla porta 80 dell'host www.microsoft.com

per inserire questa regola con ipchains dovete eseguire il seguente comando (da root)

Color xterm

```
nagash:~# ipchains -L
Chain input (policy ACCEPT):
target     prot opt     source             destination        ports
ACCEPT     tcp  ------  localnet/24        195.120.28.100     any ->   ftp
ACCEPT     tcp  ------  localnet/24        195.120.28.100     any ->   ftp-data
ACCEPT     tcp  ------  anywhere           195.120.28.100     any ->   www
DENY       tcp  ------  anywhere           195.120.28.100     any ->   0:1023

Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
nagash:~#
nagash:~#
nagash:~#
nagash:~# ipchains -L -n
Chain input (policy ACCEPT):
target     prot opt     source             destination        ports
ACCEPT     tcp  ------  192.168.1.0/24     195.120.28.100     * ->   21
ACCEPT     tcp  ------  192.168.1.0/24     195.120.28.100     * ->   20
ACCEPT     tcp  ------  0.0.0.0/0          195.120.28.100     * ->   80
DENY       tcp  ------  0.0.0.0/0          195.120.28.100     * ->   0:1023

Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
nagash:~#
nagash:~# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(f
oppy)
nagash:~#
```

Il comando ipchains -L mostro le regole correnti del firewall, lo switch -n utilizza la notazione numerica.

```
# ipchains -A output -d
                    www.microsoft.com 80 -p TCP -j DENY
```

Dopo l'inserimento di questo comando, accedere alle pagine web del sito www.microsoft.com diverrà **impossibile** (*la porta 80 TCP è proprio la porta sulla quale è in "ascolto" il demone httpd*). Bloccando tutti i pacchetti destinati a quella porta, per il vostro host sarà impossibile inoltrare la richiesta necessaria per visionare le loro pagine web. Vediamo meglio il comando appena digitato: l'opzione **-A** di ipchains indica di "appendere" la regola alla fine della lista delle regole, output specifica che la regola si riferisce ai soli pacchetti uscenti mentre l'opzione **-d** indica l'indirizzo di destinazione di questi, ovvero il campo destinazione del protocollo IP, ed è seguito da "80", che nel caso del protocollo TCP indica che la porta di destinazione è appunto la 80. L'opzione **-p** indica il protocollo dei pacchetti, ovvero il campo "protocol" dell' IP, infine l'opzione **-j** indica come comportarsi con quel dato pacchetto, in questo caso l'operazione richiesta è "DENY", ovvero il pacchetto verrà semplicemente scartato.
È da notare che quando per destinazione si utilizza il nome di un host (invece che l'indirizzo IP) saranno **aggiunte** tante regole **uguali** per tutti gli indirizzi in cui tale nome risolve. Un esempio può chiarire meglio il concetto:

```
$ nslookup www.yahoo.com
Server:    dns.tin.it
Address:   194.243.154.62

Non-authoritative answer:
Name:     www.yahoo.com
Addresses:  204.71.200.67, 204.71.200.68,
                          204.71.200.75, 204.71.200.74
```

Come è possibile osservare il nome www.yahoo.com risolve in più di un indirizzo IP. È per questo che digitare il comando:

```
# ipchains -A output -d www.yahoo.com 80
                                          -p TCP -j DENY
```

equivale a digitare i seguenti 4 comandi

```
# ipchains -A output -d 204.71.200.67 -p TCP -j DENY
# ipchains -A output -d 204.71.200.68 -p TCP -j DENY
# ipchains -A output -d 204.71.200.75 -p TCP -j DENY
# ipchains -A output -d 204.71.200.74 -p TCP -j DENY
```

La forma con il nome dell'host è preferibile: nel caso gli indirizzi ip di www.yahoo.com venissero **modificati** la riga di comando necessaria rimarrebbe immutata.

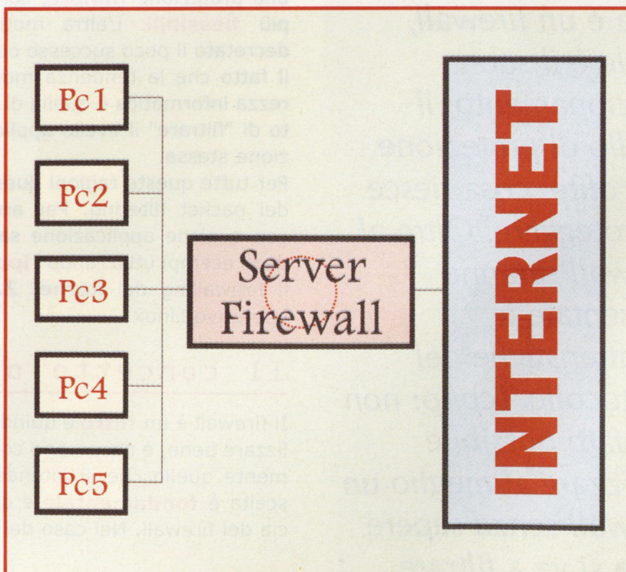
Una alternativa valida puo' essere il seguente comando:

```
# ipchains -A output -d 204.71.200.0/24 80
                                          -p TCP -j DENY
```

che indica ad ipchains di bloccare qualunque pacchetto diretto verso la porta 80 di tutti gli indirizzi IP che possiedono **almeno i 24 bit piu' a sinistra** uguali a 204.71.200.0 (ovvero gli indirizzi 204.71.200.0, 204.71.200.1, ..., 204.71.200.255).
Ciò equivale a dire che saranno bloccati tutti gli accessi verso la porta 80 di quella 'Classe C' (per informazioni sulle classi degli indirizzi IP si consiglia la lettura del NET3-HOWTO).

La relazione tra le regole

Un concetto molto importante è quello della **relazione tra le regole**. Si supponga un host su internet, che nasconde una rete locale. Questo host è per l'internet un server www, ma per la rete locale è anche un server ftp. Noi desideriamo che l'accesso all'ftp sia garantito solo dall'interno, ovvero dalla rete locale, mentre dall'esterno deve essere visibile solo l'httpd. Supponiamo che lo scenario sia quello di figura 1.
Come potete vedere PC1, PC2, ..., PC5 sono i client sulla rete locale. Anche il Server fa parte della rete locale ma ha **due** interfacce, una con un indirizzo locale e l'altra con un indiriz-





```
Color xterm
ipchains 1.3.5, 26-June-1998

Usage: ipchains -[ADC] chain rule-specification [options]
       ipchains -[RI] chain rulenum rule-specification [options]
       ipchains -D chain rulenum [options]
       ipchains -[LFZNX] [chain] [options]
       ipchains -P chain target [options]
       ipchains -M [ -L | -S ] [options]
       ipchains -h [icmp] (print this help information, or ICMP list)

Commands:
  -A chain          Append to chain
  -D chain          Delete matching rule from chain
  -D chain rulenum
                    Delete rule rulenum (1 = first) from chain
  -I chain [rulenum]
                    Insert in chain as rulenum (default = 1 = first)
  -R chain rulenum
                    Replace rule rulenum (1 = first) in chain
  -L [chain]        List the rules in a chain (or all chains)
  -F [chain]        Flush (delete all rules) chain or all chains
  -Z [chain]        Zero counters in chain (or all chains)
  -C chain          Test this packet on chain
  -N chain          Create a new user-defined chain
  -X chain          Delete a user-defined chain
  -P chain target
                    Change policy on chain to target
  -M -L             List current masqerading connections
  -M -S tcp tcpfin udp
                    Set masquerading timeout values

Options:
  -b                insert two rules: one with -s & -d reversed
  -p [!] proto      protocol (tcp, udp, icmp, all or a number)
  -s [!] address[/mask] [!] [port ...]
                    source specification
  -d [!] address[/mask] [!] [port ...]
                    destination specification
line 1/51 73%
```

Una parte dell'output del comando ipchains --help

zo pubblico. Si ammetta che la rete locale sia 192.168.1.0 e che l'indirizzo pubblico sia 195.120.28.100. Quello che serve è semplice: si desidera che da internet sia accessibile solo il web server, e che dalla rete locale sia accessibile anche il server ftp. Una soluzione potrebbe essere quella di usare le seguenti regole:

- **accetta tutti i pacchetti entranti provenienti dalla rete locale e diretti verso la porta ftp del server**
- **blocca tutti i pacchetti entranti destinati alla porta ftp del server.**

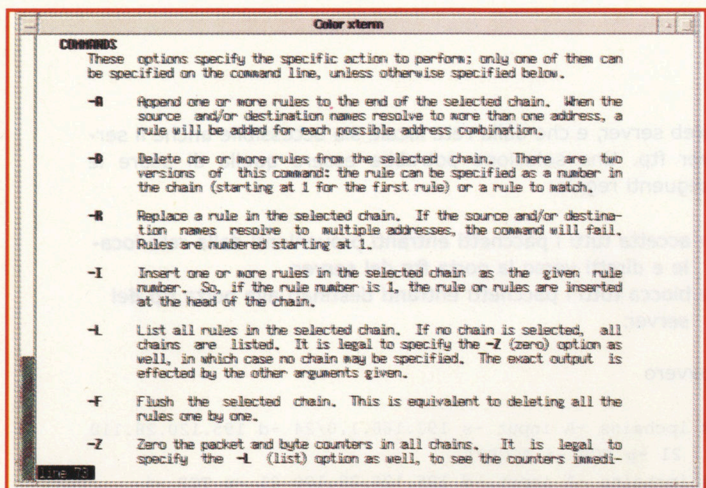
ovvero

```
# ipchains -A input -s 192.168.1.0/24 -d 195.120.28.110
  21 -p TCP -j ACCEPT
# ipchains -A input -d 195.120.28.100 21 -p TCP -j
  DENY
```

Queste due regole permettono di capire in che modo ipchains (e molti altri firewalls) processa le regole e dunque la relazione tra le regole stesse. Quando un pacchetto arriva, il firewalling di linux controlla se questo riguarda la prima regola della lista di regole inserite. Nel caso in cui il pacchetto corrispondesse, verrebbe eseguita l'operazione che la regola specifica (nel nostro caso ACCEPT), altrimenti si passerebbe alla regola successiva (questo dà l'idea di quanto il firewalling affatichi la nostra CPU, in presenza di molte regole e di elevati traffici) e così via. È per questo che le due regole sopra riportate fun-



Sicurezza di Rete



Color xterm

COMMANDS
These options specify the specific action to perform; only one of them can be specified on the command line, unless otherwise specified below.

-A Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

-D Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

-R Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

-I Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain.

-L List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the -Z (zero) option as well, in which case no chain may be specified. The exact output is effected by the other arguments given.

-F Flush the selected chain. This is equivalent to deleting all the rules one by one.

-Z Zero the packet and byte counters in all chains. It is legal to specify the -L (list) option as well, to see the counters immedi-

line 73

Una pagina del manuale di ipchains che illustra i comandi per la gestione delle regole

zionano. Immaginate un pacchetto TCP proveniente dalla rete locale e destinato alla porta 21 del server. Un tale pacchetto interesserà la prima regola, infatti l'indirizzo di provenienza rientra in 192.168.1.0/24 e quello di destinazione è proprio 195.120.28.110, come la porta di destinazione è appunto la 21, ovvero quella dell'ftp. Il kernel eseguirà l'operazione descritta dalla prima regola: ACCEPT. Il pacchetto sarà accettato e tutte le altre regole ignorate. D'altra parte un pacchetto TCP proveniente da un qualunque altro indirizzo di internet e diretto alla porta 21 del nostro server **non soddisferà** la prima regola, infatti l'indirizzo sorgente non corrisponderà (spoofing permettendo, ma è un altro discorso) con 192.168.1.0/24.

Il confronto proseguirà prendendo in esame la seconda regola, che dice di bloccare qualunque pacchetto destinato alla porta 21 del server. In questo caso il pacchetto ha tutti i requisiti richiesti, e l'operazione DENY sarà eseguita, così il pacchetto verrà scartato.

Le due regole presentate sopra non sono di certo le migliori: non danno un grosso margine di sicurezza poichè filtrano **solo** i pacchetti diretti alla porta 21. Ammettendo che per un errore il servizio telnet **non** sia stato rimosso dal server, questo continuerà ad essere accessibile da parte di internet. Un insieme di regole più "**selettive**" è il seguente:

- accetta i pacchetti TCP entranti provenienti dalla rete locale e diretti alla porta 21(ftp) e 20(ftp-data) del server
- accetta i pacchetti TCP entranti provenienti da qualunque destinazione diretti alla 80(web) del server
- blocca tutti i pacchetti TCP entranti diretti alle porte da 0 a 1024 del server (In modo da proteggere tutti gli eventuali altri servizi che "ascoltano" su una porta privilegiata).

ovvero

```
# ipchains -A input -s 192.168.1.0/24
              -d 195.120.28.100 21 -p TCP -j ACCEPT
# ipchains -A input -s 192.168.1.0/24 -d 195.120.28.100
                                20 -p TCP -j ACCEPT
# ipchains -A input -d 195.120.28.100 80
                                   -p TCP -j ACCEPT
# ipchains -A input -d 195.120.28.100 0:1023
                                   -p TCP -j DENY
```

(*n.b. omettendo -s l'indirizzo sorgente viene impostato a 0/0, dunque la regola sarà valida per gli ip con qualunque indirizzo sorgente)*

Coadiuvando questi brevi esempi con la man page di ipchains sarete capaci di sfruttare al massimo le potenzialità del firewalling del kernel 2.2, ed alla fine riuscirete a filtrare qualunque pacchetto IP, UDP, TCP ed ICMP (limiti di ipchains permettendo).

Cosa filtrare in un contesto reale

Imparare ad usare bene un determinato firewall così da essere in grado di filtrare in maniera selettiva qualunque tipo di pacchetto non basta (*ma è una condizione necessaria*) per garantirne il funzionamento ottimale: **serve sapere cosa filtrare**. Si pensi al caso specifico di un firewall "**orientato**" all'host, ovvero che protegge solo ed esclusivamente l'host su cui è installato; è il caso più semplice poichè i firewalls che proteggono intere reti sono di più difficile comprensione e meritano un articolo a parte.

Esistono anche casi molto complessi ed articolati, come le **DMZ**, che saranno trattate in futuro su queste pagine. Come dicevamo l'esempio riguarda il classico host su internet che è fisicamente il computer su cui girano i servizi di rete ed anche quello su cui gira il firewall stesso.

Immaginiamo che su questo ipotetico host siano presenti soltanto il servizio di ftp anonimo ed il server dns. La **scelta dei demoni** è importante almeno quanto il **firewalling**: in questo caso sarebbe consigliabile installare un ftp che permette solo l'accesso anonimo, ne esiste uno molto buono di Marcus J. Ranum. Per il server dns useremo certamente il classico bind. Le nostre regole dovranno permettere l'accesso da parte di tutta internet verso la porta 20 e 21 del protocollo **TCP** e verso la porta 53 del protocollo **UDP** dell'host.

Ipotizzando che il nostro host non abbia altro scopo oltre quello di far girare questi due demoni potremmo pensare alla seguenti regole:

1a - bloccare in entrata tutti i pacchetti con indirizzo ip sorgente 127.0.0.0/8
1b - bloccare in entrata tutti i pacchetti con indirizzo ip sorgente uguale al nostro ip.
1c - bloccare in entrata tutti i pacchetti con indirizzo ip sorgente uguale a 192.168.1.0/24

2a - accettare in entrata i pacchetti TCP verso la porta 21
2b - accettare in entrata i pacchetti TCP verso la porta 20
2c - bloccare in entrata i pacchetti TCP verso le porte da 0 a 1023
2d - bloccare in entrata i pacchetti TCP con il solo flag SYN settato verso le porte da 1024 a 65535

3a - accettare in entrata i pacchetti UDP verso la porta 53



3b - bloccare in entrata i pacchetti UDP verso le porte da 0 a 1023

4a - permettere in entrata i pacchetti ICMP di tipo 3, 4 e 11
4b - bloccare in entrata tutti i pacchetti ICMP
4c - permettere in uscita i pacchetti ICMP di tipo 4
4d - bloccare in uscita tutti i pacchetti ICMP

N.B. Esaminando queste regole non dimenticate il modo in cui sono processate!
Vediamo di capire il perchè di queste regole. Le regole 1a e 1b servono a prevenire lo **spoofing** dei pacchetti: non c'è alcun motivo di accettare pacchetti provenienti dall'indirizzo di **loopback**, dal nostro stesso ip o dagli ip della nostra rete locale (*se mai ce ne fosse una*). Ricordate che le seguenti regole devono essere specificate utilizzando l'opzione **-i** di ipchains specificando l'interfaccia con cui siamo connessi ad internet. Infatti non specificando l'interfaccia queste regole avrebbero l'effetto indesiderato di **non permettere** le connessioni originate da quegli ip neppure nel caso fossero leggittime. Le regole 2a e 2b permettono le connessioni verso le porte del servizio ftp, mentre le regole 2c e 2d bloccano rispettivamente ogni tipo di accesso alle porte TCP privilegiate (da 0 a 1023) e l'inizio di una connessione verso le porte non privilegiate da 1024 a 65535 (SYN flag settato).
Una piccola nota, la regola 2d costringerà ad usare i **clients ftp** dell'host in **passive mode**. La regola 3a permette l'accesso al server dns, mentre la regola 3b blocca l'accesso a tutte le porte UDP privilegiate. Le regole 4a e 4b si occupano di permettere l'accesso ai soli pacchetti ICMP di tipo destination **unreachable**, **source quench** e **time exceeded**: tranne rari casi è utile non filtrare questi pacchetti almeno in entrata. Infine le regole 4c e 4d permettono in uscita solo gli ICMP di tipo **source quench**. Qualcuno potrebbe non essere d'accordo sul fatto di bloccare il tipo 3 in uscita. In effetti bisogna valutare di volta in volta. Per un host su cui girano solo ftp e dns mi pare una buona scelta, ed è un modo efficace per bloccare gli scan UDP.

Conclusione

Scegliere adeguate regole di firewalling non è una cosa **semplice** perchè non è un processo **schematico**. Ogni caso è diverso dall'altro e si deve scegliere un compromesso tra **flessibilità** e **sicurezza**. Non ci sono regole di firewalling adeguate se chi le implementa non conosce in maniera approfondita il TCP/IP ed i protocolli dei servizi che girano sull'host. Nel caso di firewalls che proteggono intere reti le cose sono più complesse. Solo una attenta valutazione e dei test accurati possono portare ad una configurazione che aumenta realmente la sicurezza globale della rete protetta.





INSECURITY NEWS



a cura di antirez

A distanza di molti mesi dall'ultimo numero è uscito Phrack 55, la nota e-zine è disponibile alla url http://www.phrack.org Tra gli argomenti trattati sul questo numero ci sono un rootkit per NT creato facendo reverse engeenering, un tributo al grande Richard Stevens, un articolo sulle vulnerabilità dei CGI in perl e molto altro.
Altra news interessante: è stata resa pubblica la nuova patch di Solaris Designer per il kernel 2.2.12 di linux, la patch, che rende più sicure varie parti del kernel è disponibile alla url: http://www.openwall.com/linux.

Programma: sshd

Versione: 1.2.26 e 1.2.27 (probabilmente anche le versioni precedenti).
Sistemi vulnerabili: tutti (è stato testato solo su Linux).
Data: 3 Settembre.
Impatto: Utilizzando uno script simile a "`while true; do telnet host port & done`" è possibile rendere la macchina inutilizzabile da remoto.
Soluzione: Aspettare la patch, altrimenti wrapperare ssh utilizzando xinetd con una configurazione simile alla seguente:
service ssh

```
{
    socket_type         = stream
    protocol            = tcp
    wait                = no
    user                = root
    server              = /usr/local/sbin/sshd
    server_args         = -i
    instances           = 10
}
```

Programma: Netscape Communicator

Versione: 4.06J, 4.5J-4.6J, 4.61e (forse anche la 3.0)
Sistemi vulnerabili: almeno Windows 95/98
Data: 2 Settembre.
Impatto: Netscape communicator nelle versioni specificate contiene un buffer overflow remoto exploitabile, questo significa che un visitando un sito che contiene delle pagine create ad-hoc è possibile far eseguire del codice arbitrario al vostro browser! Ciò è molto grave e vi consiglio di upgradare immediatamente il browser all'ultima versione disponibile. Lo scopritore della vulnerabilità ha creato della pagine che permettono di testare la vulnerabilità.
Attenzione, visitatele a vostro rischio e pericolo! le URLs sono le seguenti:
http://www.ugtop.com/defcon0/hc/nc4x_ex/nc4x_ex.cgi
http://www.ugtop.com/defcon0/hc/nc4x_ex/nc4x_ex2.cgi
Soluzione: aggiornare il vostro browser all'ultima versione disponibile.

Programma: Proftpd

Versione: 1.2.0pre6 e precedenti.
Sistemi vulnerabili: tutti i sistemi unix che lo utilizzano.
Data: 16 Settembre.
Impatto: Sono stati scoperti vari buffer overflow nel demone Proftpd ed è stato verificato che la vulnerabilità è exploitabile, ovvero che è possibile eseguire comandi con i privilegi di root sul sistema remoto che utilizza le versioni di Proftpd vulnerabili.
Soluzione: Proftpd pare essere un software profondamente insicuro, la migliore soluzione è utilizzare un demone ftpd diverso, almeno fino a quando non siano rilasciate un paio di nuove versioni e ne sia verificata la sicurezza.
Nel frattempo se vi serve un demone ftp **soltanto anonimo** vi consiglio vivamente di utilizzare anonftpd, di Bernstein che si può scaricare dal seguente indirizzo: ftp://koobera.math.uic.edu/www/anonftpd.html
Altrimenti se non vi servono particolari features un buon ftp normale è quello di OpenBSD, disponibile anche per Linux.

Programma: Lynx

Versione: 2.8.2 e precedenti.
Sistemi vulnerabili: tutti i sistemi su cui gira.
Data: 16 Settembre.
Impatto: Lynx, il famoso browser solo testo, quando esegue programmi esterni, come ad esempio il telnet non filtra la stringa passata per eseguire il programma.
Questo permette di inserire dei parametri di linea di comando. Per esempio del codice HTML come ad esempio `<A HREF="telnet://-n.rhosts"> click me</a>` può creare o sovrascrivere in file .rhosts. Questa vulnerabilità è più o meno grave a seconda dei clients che il vosto lynx permette di eseguire, ovvero dalla vostra configurazione.
Per esempio nel caso in cui ssh o rsh fossero permessi è possibile eseguire comandi remoti sul sistema vulnerabile.
Soluzione: Aggiornare lynx all'ultima versione o disabilitare l'esecuzione dei programmi esterni (ma è consigliato aggiornare).

Programma: /usr/bin/pb ed /usr/bin/pg

Sistemi vulnerabili: Linux/SuSE.
Data: 16 Settembre.
Impatto: I programmi pb e pg, che su SuSE sono setuid root di default, permettono a qualunque utente locale di leggere

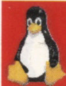
E' possibile reperire le patch alle seguenti URL:

Debian	http://www.debian.org/security/
RedHat	http://www.redhat.com/corp/support/errata/
SuSE	http://www.suse.de/security/
Caldera	http://www.calderasystems.com/news/security/

In tutti gli altri casi e' necessario scaricare l'ultima versione della applicazione dal relativo sito.

Per quanto l'articolo sia redatto con la massima cura, nè «Linux&C.» nè l'autore potranno ritenersi responsabili degli usi e/o abusi delle informazioni riportate.

InSicurezza: bugs & patches



qualsiasi file del sistema per controllare se siete vulnerabili fate un link simbolico da pb.conf a /etc/shadow ed eseguite il programma pb. Dovreste vedere tutte le righe dei file shadow segnate da pb come "Unknown config line". Questi programmi contentogo altre insicurezza, come buffer overflow e procedure di lettura/scrittura dei file non sicure.
Soluzione: togliere il setuid bit ai due programmi.

Programma: amd (Berkeley Automounter Daemon)

Sistemi vulnerabili: tutti i sistemi che utilizzano amd.
Impatto: c'è un buffer overflow nel codice di logging del demone, attacker remoti, sfruttando questa vulnerabilità, possono eseguire codice arbitrario con gli stessi permessi del demone (usualmente il demone gira da root).
Soluzione: upgradare amd all'ultima versione disponibile.

Programma: sccw (SuSE 6.2)

Sistemi vulnerabili: Linux/SuSE 6.2.
Data: 16 Settembre.
Impatto: Il programma /usr/bin/sccw, che nella SuSE 6.2 è suid root per default permette di leggere qualunque file sul sistema con i privilegi di root, dunque sarà possibile leggere anche /etc/shadow ed altri file 'sensitivè.
Basta eseguire il programma, scegliere l'opzione '8' e specificare il file che si vuole visualizzare.La visualizzazione è tutta in maiuscolo senza i segni di punteggiatura, dunque non è detto che sia semplice crackare le password. Una cosa molto semplice per cui questo bug potrebbe essere usato è quella di leggere la posta degli altri utenti del sistema.
Un'altro grave problema è un buffer over flow nella gestione della variabile HOME. Come se non bastasse il programma segue i links simbolici del file .sccw.
Questo programma setuid, che contiene tre vulnerabilità classiche in una sola piccola applicazione, da l'idea di come un `find / -perm -4000` possa essere uno strumento di hardening non da poco.
Soluzione: `chmod -s /usr/bin/sccw`

Programma: kernel di Solaris 2.6

Sistemi vulnerabili: Solaris 2.6.
Data: 18 Settembre.
Impatto: Eseguire il programma nmap versione 2.08 utilizzando l'opzione per l'OS fingerprint (-O) verso una porta su cui è in ascolto un particolare demone TCP può creare un kernel panic non appena tale demone viene killato.
Soluzione: installare la patch 105529-07 (per sparc) o la 105530 (per i386).

Programma: Mutt

Versione: tutte le versioni precedenti la 1.0pre3.
Sistemi vulnerabili: tutti.
Data: 18 Settembre.
Impatto: L'ottimo mail client mutt ha un buffer overflow nella gestione del text/enriched che potrebbe essere exploitato tramite un messaggio email appositamente formattato. Questo significa che un attacker potrebbe eseguire del codice arbitrariamente scelto sulla macchina attaccata semplicemente spedendo una mail.
Soluzione: upgradare immediatamente alla versione 1.0pre3.

Programma: Internet Explorer

Versione: 5.0
Sistemi vulnerabili: Windows 95, NT 4.0, probabilmente 98.
Data: 27 Settembre.
Impatto: è possibile leggere file locali. La vulnerabilità può essere exploitata utilizzando del messaggi email o dei messaggi dei newsgroup in formato HTML. è disponibile una dimostrazione dell'exploit alla URL http://www.nat.bg/~joro/download2.html
Soluzione: Disabilitare l' "Active Scripting".

Programma: Kernel di Linux

Versione: 2.2.x.
Sistemi vulnerabili: tutti i sistemi linux 2.2.x
Impatto: La chiamata di sistema bind(2) eseguita su un socket di dominio di Unix segue i links simbolici, mentre in molti altri sistemi unix la chiamata fallisce con l'errore EADDRINUSE. Ciò può causare dei problemi con alcuni programmi, tra i quali ssh: è possibile creare un socket di dominio di unix in qualunque parte nel filesystem.
Soluzione: Ancora non disponibile.

Programma: kvt

Sistemi vulnerabili: tutti i sistemi con la versione di KDE che contiene kvt.
Data: 29 Settembre 1999.
Impatto: Se un attacker riesce a far visualizzare in qualunque modo una determinata stringa di caratteri sul kvt di un utente vulnerabile può exploitare un buffer overflow eseguendo codice arbitrario.
Come l'xterm kvt ha una feature che permette tramite una sequenza di escape particolare di cambiare il titolo alla finestra. Con kvt ciò è ancora possibile ma non c'è il controllo della lunghezza del buffer, così se la vittima visualizza con cat un file appositamente formattato speditogli dall'attacker o se tramite il client ftp visualizza un banner appositamente pensato o in qualunque altro modo visualizza questa stringa da kvt tale buffer overflow può essere exploitato. La versione di KDE presente con RedHat 6.0 è vulnerabile.
Soluzione: passare da kvt a kconsole, presente nelle nuove versioni di KDE.

Programma: cfingerd

Versione: 1.4.2 e precedenti
Sistemi vulnerabili: tutti quelli che non limitano la lunghezza del gecos.
Data: 21 Settembre.
Impatto: Un utente locale può eseguire codice aritrario con gli stessi privilegi dell'owner di cfingerd (di solito root o nobody) nei sistemi che non limitano la lunghezza del gecos utilizzando un gecos studiato appositamente.
L'exploit è dunque locale. Un esempio di sistema che non limita la lunghezza del gecos è FreeBSD per cui è disponibile un esempio di exploit su internet.
Soluzione: Utilizzare un demone finger sicuro.



Salvatore Sanfilippo è esperto di sicurezza delle reti, si occupa di programmazione distribuita con il protocollo Tcp/Ip, sviluppo di software per il controllo e la sicurezza. Attualmente collabora con Alicom Italia (www.alicom.com). È raggiungibile all'indirizzo email s.sanfilippo@oltrelinux.com; la sua home page è http://www.kyuzz.org/antirez

Seconda Puntata

il nostro DATABASE sul World Wide Web

FILIPPO STEFANELLI
f.stefanelli@oltrelinux.com

WebMaster di dislessici.org si occupa di sicurezza di siti web e del loro interfacciamento con basi di dati.

Portare un database su internet è una esigenza ormai sentita da molti: la scorsa puntata abbiamo fatto il punto su cosa serve per automatizzare un sito Web. Ecco le prime funzioni di PHP che permettono una certa interazione con il database. Presto sarete in grado di costruire la Vostra personale applicazione.

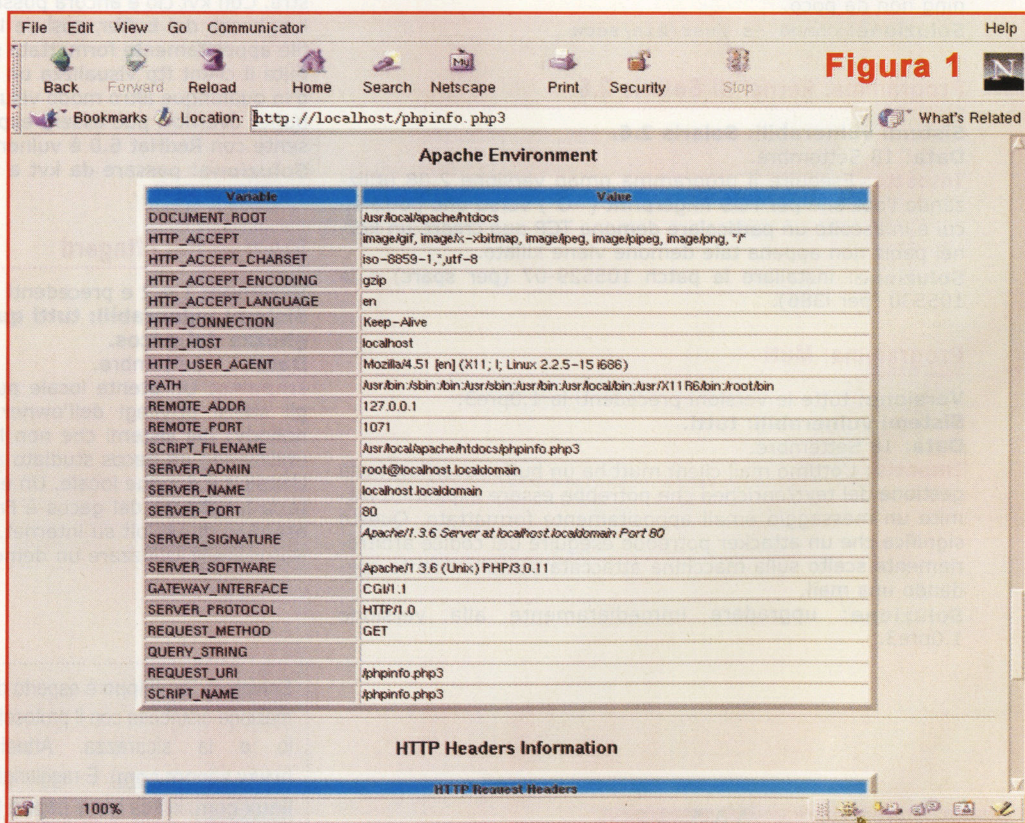
In questa puntata verrà spiegato come realizzare una pagina che sfrutti le risorse offerte da PHP e come manipolare dei dati all' interno di un database MySQL. Nella precedente puntata abbiamo visto alla fine un piccolo esempio di un' istruzione in PHP, ora è il momento di sfruttarlo appieno all' interno delle nostre pagine.

PHP all' opera

Iniziamo guardando una panoramica generale di PHP all' interno del nostro computer. Con il comando **phpinfo**() ci viene restituita una lunga lista di informazioni che riguardano tutti i settaggi attuali di PHP. Proviamo quindi ad inserire questo comando all' **interno** di una pagina html e, se PHP è stato installato correttamente, una volta indirizzato il browser verso questa pagina verrà visualizzata una lista di impostazioni simili a quelle della Figura 1.

Esaminando il Riquadro 1 analizziamo ora un esempio più vicino a una pagina web vera e propria. Dopo aver copiato il sorgente in un file di nome primoesempio.php3, proviamo ad aprirlo con un browser. Il risultato sarà molto simile a quello che è illustrato nella Figura 2.

Realizzando questo esempio si è cercato di utilizzare in poco spazio molti comandi PHP, spesso sono mostrate più **soluzioni** per eseguire la stessa operazione, chi conosce già PHP troverà quindi delle ridondanze e un modo non molto "elegante" di risolvere alcuni problemi; in ogni caso ritengo che questo sia un metodo molto efficace per apprendere



Variable	Value
DOCUMENT_ROOT	/usr/local/apache/htdocs
HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
HTTP_ACCEPT_CHARSET	iso-8859-1,*,utf-8
HTTP_ACCEPT_ENCODING	gzip
HTTP_ACCEPT_LANGUAGE	en
HTTP_CONNECTION	Keep-Alive
HTTP_HOST	localhost
HTTP_USER_AGENT	Mozilla/4.51 [en] (X11; I; Linux 2.2.5-15 i686)
PATH	/usr/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/X11R6/bin:/root/bin
REMOTE_ADDR	127.0.0.1
REMOTE_PORT	1071
SCRIPT_FILENAME	/usr/local/apache/htdocs/phpinfo.php3
SERVER_ADMIN	root@localhost.localdomain
SERVER_NAME	localhost.localdomain
SERVER_PORT	80
SERVER_SIGNATURE	*Apache/1.3.6 Server at localhost.localdomain Port 80*
SERVER_SOFTWARE	Apache/1.3.6 (Unix) PHP/3.0.11
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.0
REQUEST_METHOD	GET
QUERY_STRING	
REQUEST_URI	/phpinfo.php3
SCRIPT_NAME	/phpinfo.php3

Figura 1



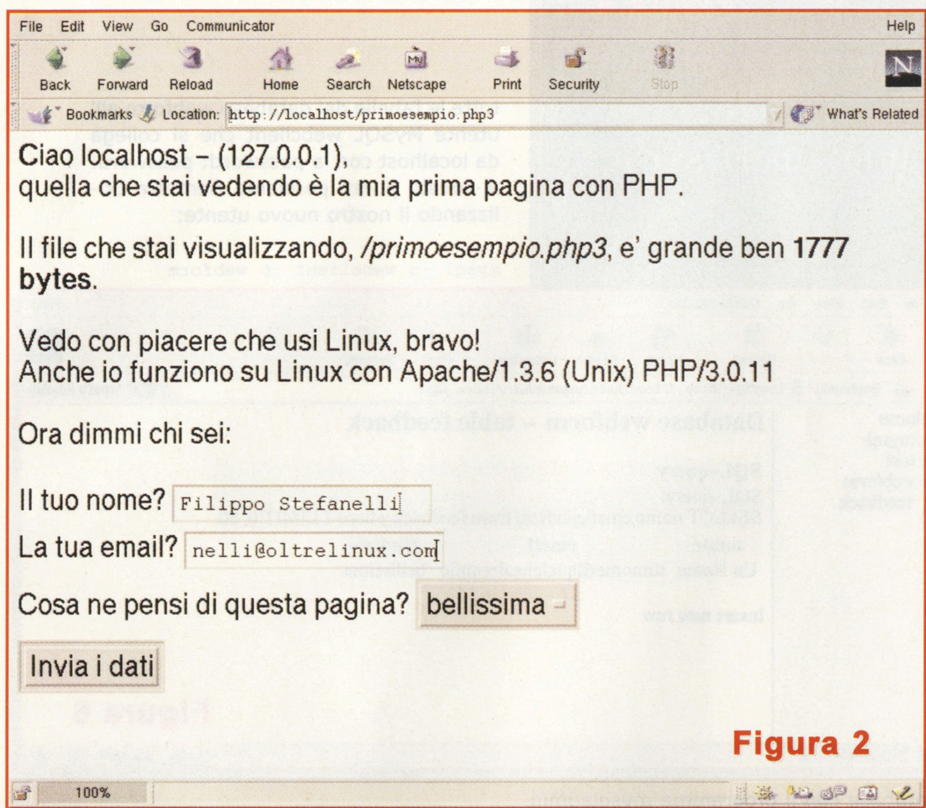


Figura 2

PHP da zero. Passiamo adesso ad analizzare il sorgente nel dettaglio.
Da una prima rapida lettura si può facilmente vedere come il codice PHP possa essere inserito **veramente ovunque** all' interno di una pagina, l' importante è che sia sempre delimitato dalle **tag** di **apertura** e **chiusura** che abbiamo descritto nella precedente puntata. Alla terza riga incontriamo la nostra prima funzione: **echo**() (*non è propriamente una funzione ma questo non ci interessa ora*). Questa permette di **visualizzare** una stringa di caratteri. Nel nostro caso echo() fa scrivere la parola PHP unita al risultato della funzione **phpversion**() (*funzione che restituisce il numero della versione di PHP*). Fra la parola che vogliamo visualizzare e la funzione c'è un operatore di PHP: un "**.**" che serve a unire degli elementi tra di loro (*operatore di concatenazione*). Tralasciamo per il momento il primo blocco del ciclo if e incominciamo ad analizzare il sorgente sottostante.
Le prima novità che incontriamo sono delle variabili. Queste generalmente sono contraddistinte da un "**$**" scritto prima del nome. In questo caso la variabile **$HTTP_HOST** è una variabile che viene passata a PHP dal server web, Apache. Questi tipi particolari di variabili sono elencati dal comando phpinfo() che abbiamo visto prima. In questo primo esempio ci sono numerose variabili di sistema, anche queste sono tutte elencate da phpinfo().
Continuando ad analizzare il codice arriviamo a **$misura_file** = filesize(__FILE__). Qui assegniamo ad una variabile creata da noi un valore, in questo caso il risultato della funzione filesize(), e poi utilizziamo echo per visualizzare il contenuto della nostra nuova variabile. In questa riga c'è una particolarità, l' argomento della funzione filesize è "__FILE__". Questa è una "costante" di PHP. Una costante è una identificativo che, una volta definita, non può più essere **cambiata**. Vi sono alcune costanti predefinite di PHP che sono elencate nel paragrafo sette del secondo capitolo del manuale. Per definire costanti aggiuntive si usa la funzione **define**().
Tornando al nostro codice troviamo ora dei commenti. Vi sono diverse modalità per fare un commento, PHP ha ereditato i commenti in stile C/C++ che vengono delimitati da "/*" e "*/" e anche i commenti su linea singola introdotti da "//". Entrambi si possono vedere nel nostro esempio. Esiste anche un terzo sistema per fare commenti all' interno del codice PHP e cioè attraverso un "#" (*come all'interno di una shell*).
Indipendentemente dal modello che si vuole utilizzare, commentare i programmi aiuta moltissimo gli altri a capire il sorgente più facilmente e noi a fare delle modifiche future. Continuando nell' analisi del sorgente, dopo aver definito la variabile $best_os, troviamo un istruzione di controllo di flusso: if. Anche senza nessuna precedente esperienza di programmazione è abbastanza intuitivo il funzionamento dell' istruzione if. Il comportamento è il seguente: all' inizio viene controllata la condizione tra parentesi, in questo caso "$OSTYPE == $best_os", se questa risulta non nulla allora si procede eseguendo il codice contenuto nel corpo di if (quello racchiuso da "{"e "}"), altrimenti il ciclo si blocca, oppure, se è presente, vengono eseguite le istruzioni contenute nel corpo dell' else. Formalmente l' istruzione if può anche essere scritta in un altro modo, descritto nel manuale di PHP al

```
Subject: Invio form dal web 12-08-99
From: webserver@localhost.localdomain
Reply-To: webmaster@localhost.localdomain

Sono arrivati dei dati da 127.0.0.1 alle 9:56 pm.

Filippo Stefanelli ha valutato la pagina: bellissima.

Se lo vuoi contattare la sue email e' f.stefanelli@oltrelinux.com.


Filippo Stefanelli ha visto la pagina con Mozilla/4.51 [en] (X11; I;
Linux 2.2.5-15 i686).

:
```
Figura 3



WEB Server

```
mysql>
mysql> SELECT * FROM feedback;
+---------+-------------------------+-------------+
| nome    | email                   | giudizio    |
+---------+-------------------------+-------------+
| Un Nome | unnome@qualche.dominio  | bellissima  |
+---------+-------------------------+-------------+
1 row in set (0.31 sec)

mysql>
```

Figura 4

decimo paragrafo del secondo capitolo. Osserviamo ora la condizione di if: "$OSTYPE == $best_os". L' operatore che relaziona i due argomenti è "==" e significa "uguale a". Vi sono altri operatori per definire tutti i possibili casi e sono: "!=",">","<",">=","<=". Il loro significato è illustrato nel secondo capitolo del manuale al paragrafo nove. Analizzando il corpo di if vediamo la funzione **print**(), alternativa a echo() e anche quelle variabili di cui abbiamo già parlato che sono passate a PHP dal server: $SERVER_SOFTWARE e qualche riga prima $OSTYPE. Alla fine del nostro primo esempio troviamo una form. L' action della form è la pagina stessa quindi, una volta premuto il pulsante di invio, la form viene processata dal codice presente all' interno di questa pagina. Il codice in questione è proprio quel blocco if che abbiamo evitato di analizzare all' inizio. In che modo if capisce di dover **analizzare la form**? Cercando l'esistenza della variabile **$submit**. Infatti tutti i valori dei campi di una form sono inseriti in delle variabili che vengono nominate in base al valore di "name" della form. Nel nostro caso le variabili che vengono passate a PHP dalla form sono: $nome, $email, $giudizio e anche $submit. Abbiamo aggiunto nella form il campo name="submit" per dichiarare questa variabile e quindi poterne controllare poi la sua esistenza attraverso il ciclo if all' inizio dello script. Semplice ed efficace. Il corpo dell' if non fa altro che mandarci una mail con tutti questi parametri come si può vedere dalla Figura 3. Come abbiamo visto la pagina in questione non è assolutamente difficile da realizzare e offre molti spunti interessanti per un sito web, ora è il momento della parte più dinamica e interattiva.

Iniziamo con MySQL

Per lavorare con MySQL dobbiamo creare un database per contenere tutte le tabelle che ci serviranno. Creiamo il database con il programma mysqladmin che si trova nella directory bin dell' installazione di mysql. Andiamo nella directory dove abbiamo installato MySQL, nel nostro caso /usr/local/mysql e lanciamo:

```
bin/mysqladmin create webform
```

Ora è il momento di creare un utente speciale che possa accedere solo nel database webform. Questo è il metodo migliore per non compromettere la sicurezza e l'integrità dei nostri dati all'interno dei database di MySQL. Per far questo entriamo come **root** con il client mysql nel database webform:

```
mysql -u root webform
```

una volta dentro digitiamo il testo che segue:

```
GRANT SELECT,INSERT,UPDATE,DELETE,
                        CREATE,DROP
 ON webform.*
 TO webclient@localhost
 IDENTIFIED BY 'password';
```

Da notare il ";" che indica la fine di una qualsiasi istruzione. Grazie a questo comando abilitiamo la possibilita di eseguire select,insert,etc... all' interno di tutte le tabelle del database webform all' utente MySQL webclient che si collega da localhost con la password: password. A questo punto possiamo rientrare utilizzando il nostro nuovo utente:

```
mysql -u webclient -p webform
```

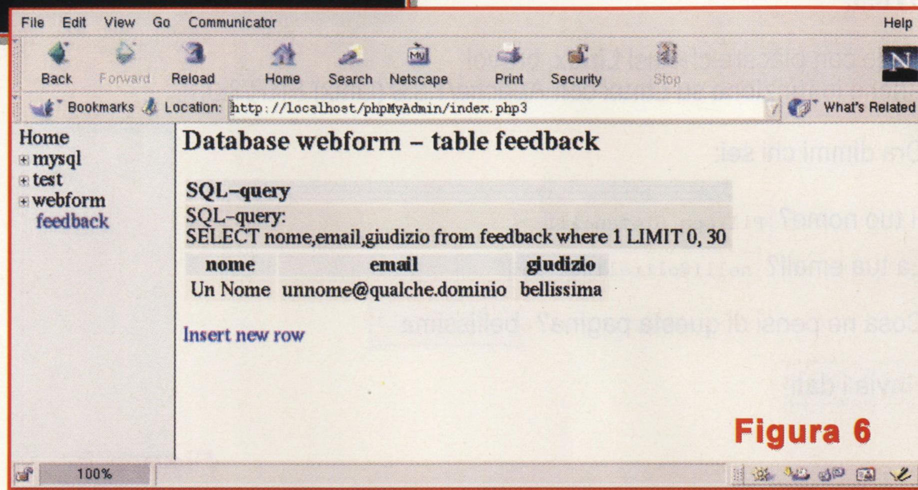


Figura 6

Ci verrà richiesta una password e dopo averla inserita, se tutto è stato fatto correttamente, possiamo iniziare a creare le tabella che ci serve all' interno del database webform. Alle colonne della nostra tabella assegniamo, per comodità, gli stessi nomi delle variabili della form.

```
CREATE TABLE feedback (
 nome CHAR(80),
 email CHAR(80),
 giudizio VARCHAR(10)
);
```

CHAR e VARCHAR identificano il tipo di dato che verrà inserito nelle rispettive colonne, fra parentesi il suo valore massimo. Proviamo ora a vedere cosa c'è dentro la nostra nuova form con questa query:

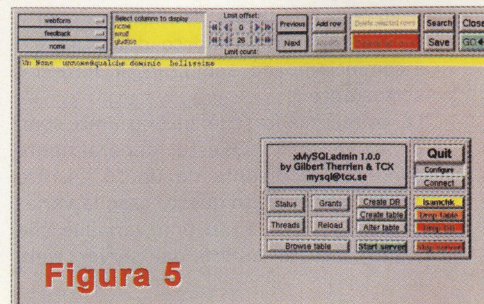


Figura 5

Linux&C. - Mensile - Release 0.1.0 - Novembre



```
SELECT * FROM feedback;
```

Naturalmente MySQL ci informerà che non vi è assolutamente nulla quindi riempiamola noi con dei dati inventati:

```
INSERT INTO feedback VALUES(
'Un
Nome','unnome@qualche.dominio','bell
issima'
);
```

Ora proviamo a rifare il SELECT di prima e vedremo comparire il nostro primo record come illustrato dalla Figura 4. Le query appena eseguite possono essere inoltrate anche attraverso dei client grafici (Figura 5) che, graficamente appunto, si interfacciano con MySQL; tutti questi programmi sono elencati nel sito di MySQL. Vi sono anche delle interfaccie web per MySQL. Una di queste, phpMyAdmin, è realizzata completamente in PHP e proprio per questo sua portabilità è uno strumento ideale per amministrare MySQL all' interno di una intranet. Il SELECT di prima, all' interno di phpMyAdmin, è illustrato dalla Figura 6. Come si può facilmente vedere esistono molti modi per lavorare con MySQL, noi però vogliamo farlo in modo trasparente da una pagina web.

Conclusioni

Per fare con PHP la query che abbiamo visto con i diversi client basta utilizzare la funzione **mysql_query**(). Assegnamo ad una variabile il contenuto della query:

```
$query_sql = "SELECT * FROM
feedback";
```

e poi inviamo la query al server MySQL memorizzando il risultato in una variabile:

```
$risultato =
mysql_query($query_sql);
```

Naturalmente prima di poter fare qualsiasi query dobbiamo dire a PHP verso quale server ci stiamo collegando, con che utente, con quale password e a quale database. Vedremo tutto questo e come realizzare un bell' output del risultato di MySQL nella prossima puntata. Inoltre creeremo dei **controlli** per i **dati** che vengono inviati dalla form in modo da aumentarne la **sicurezza** (sempre in primo piano in questo genere di applicazioni) e l'efficenza. In questo modo renderemo la nostra pagina ancora più dinamica.

Riquadro 1

```
<html>
<head>
<title> Il primo esempio con <?php echo("PHP " . phpversion()); ?></title>
</head>
<body bgcolor=#ffffff>

<?php
if ($submit) {

mail("root@localhost",
"Invio form dal web " . date("d-m-y"),
"Sono arrivati dei dati da $REMOTE_ADDR alle " . date( "g:i a" ) . ".
\n\n$nome ha valutato la pagina: $giudizio.\n\n Se lo vuoi contattare la
sue email e' $email.\n\n\n $nome ha visto la pagina con $HTTP_USER_AGENT.",
"From: webserver@localhost");

echo ("I dati sono stati ricevuti, grazie.");

} else {

?>

<font size=+2><strong>C</strong></font>iao <?php echo("$HTTP_HOST -
($REMOTE_ADDR)"); ?>, <br>

quella che stai vedendo è la mia prima pagina con PHP.<p>

Il file che stai visualizzando, <I><?php echo($SCRIPT_NAME); ?></I>,
 e' grande ben <STRONG>

<?php
$misura_file = filesize(__FILE__); echo($misura_file);
?> bytes</STRONG>.<P>

<?php

/* Ora lo salutiamo in modo diverso a seconda se usa Linux o no,
naturalmente non si offendano i non linuxiani :) */

$best_os = Linux; // Qualche define prima di incominciare.
if ($OSTYPE == $best_os) {

print("Vedo con piacere che usi $best_os, bravo!<BR> Anche io funziono su
$best_os con $SERVER_SOFTWARE<p>\n");

} else {

echo("Mmm.. non stai usando $best_os, che ne dici di cambiare?<P>\n");

}
?>

<P>
Ora dimmi chi sei:<br>
<form method="post" action="primoesempio.php3">
Il tuo nome? <input type="text" name="nome"><br>
La tua email? <input type="text" name="email"><br>
Cosa ne pensi di questa pagina? <select name="giudizio">
<option value="bellissima" selected>bellissima</option> <option
value="carina">carina</option>
<option value="orrenda">orrenda</option></select><br>
<input type="submit" name="submit" value="Invia i dati"><br>
</form>

<?php

} // fine primo if

?>

</body>
</html>
```

Bibliografia

[1] "PHP3 Manual", PHP Documentation Group.

[2] "MySQL Reference Manual for version 3.22.25", www.mysql.com





SQL

corso di base

FRANCESCO CONSUMI
f.consumi@oltrelinux.com

Si occupa di programmazione, di reti Windows e Linux e di siti Internet. Svolge attività di consulenza con varie aziende ed enti pubblici.

La gestione delle informazioni contenute nei database necessità di un linguaggio che riesca, con una sintassi pulita e rigorosa, a compiere interrogazioni senza essere limitati da estensioni proprietarie. SQL è il linguaggio principe per il trattamento dei dati, ed assicura massima portabilità.

L'utilizzo preferito di un server Linux è probabilmente quello di gestire un sito Web, e lo scopo di un sito è spesso quello di pubblicare dati; sarà difatti un argomento che ritroveremo spesso su queste pagine. Ecco quindi che ci troviamo in molte situazioni alle prese con i database. Quindi, a partire da questo numero daremo un'occhiata alla strutturazione dei dati e al più diffuso linguaggio utilizzato per gestirli: SQL.

I Database

Innanzitutto vediamo a grandi linee la composizione di un database: supponiamo di avere un elenco clienti e uno delle relative fatture: si hanno nel database almeno **tre** tabelle, i clienti, le fatture e le righe (articoli) delle fatture. Ogni tabella è costituita da diversi campi, ovvero le colonne, che nel caso dei clienti possono essere Codice, Ragione sociale, Indirizzo, etc.; i dati inseriti nelle tabelle, cioè le righe con i dati di ogni singolo cliente, formano i records (vedi figura 1).
Una volta definiti i dati, si pone il problema di ritrovarli rapidamente all' interno delle tabelle; a questo scopo si utilizzano le **chiavi** e gli **indici**: le chiavi utilizzano campi, uno solo od una combinazione di più, e sono definite in modo da essere uniche all'interno della tabella, e quindi in grado di identificare univocamente un record. Nel nostro esempio dei clienti può esserlo il **codice**, mentre non sarebbe adatta la ragione sociale, vista la possibilità di **omonimia** tra clienti. Per cercare i records secondo criteri specifici senza dover scandire tutta la tabella si utilizzano gli **indici**, che sono delle tabelle che contengono il solo campo da indicizzare e la posizione del record all'interno della tabella, ordinate secondo il campo indicizzato. Generalmente per ogni tabella si ha un **indice** associato alla chiave primaria (il codice del cliente), più altri indici secondari utilizzati per sveltire le ricerche più frequenti: nel nostro caso è quasi d'obbligo avere un indice associato alla ragione sociale, visto che è la ricerca più comune effettuata su una tabella di clienti.
Infine, nei cosiddetti database relazionali, abbiamo la possibilità di definire delle **relazioni tra tabelle**, ovvero dei legami tra records inseriti in tabelle diverse che permettono di evitare le ripetizioni dei dati; il legame viene realizzato attraverso un campo chiave presente in entrambe le tabelle: ad esempio, nella tabelle delle fatture non si riportano tutti i dati del cliente, ma solo il codice, che permette di ritrovare i restanti dati nella tabella clienti. Le relazioni possono essere di tre tipi: **uno a uno**, **uno a molti**, **molti a molti** (vedi figura 2).
Nel primo caso ad un record nella prima tabella corrisponde un solo altro record nella seconda , ovvero i due campi sono entrambi chiavi primarie o sono comunque univoci all'interno della tabella.
È un tipo di relazione raramente usato, perché generalmente si includono tutti i campi in una sola tabella: ha senso solo per spezzare tabelle con molti campi o per definire campi che servono solo per un sottoinsieme di clienti, ad esempio per memorizzare l'indirizzo di e-mail dei soli clienti che ce l'hanno.

Definition | Data | Enter SQL

```
select *
from customer
```

CustNo	Company	Addr1	Addr2	City	State	Zip	Country
1221	Kauai Dive Shoppe	4-976 Sugarloaf Hwy	Suite 103	Kapaa Kauai	HI	94766-1234	US
1231	Unisco	PO Box Z-547		Freeport			Bahamas
1351	Sight Diver	1 Neptune Lane		Kato Paphos			Cyprus
1354	Cayman Divers World Unlimited	PO Box 541		Grand Cayman			British West Indies
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj		Christiansted	St. Croix	00820	US Virgin Islands
1380	Blue Jack Aqua Center	23-738 Paddington Lane	Suite 310	Waipahu	HI	99776	US
1384	VIP Divers Club	32 Main St.		Christiansted	St. Croix	02800	US Virgin Islands
1510	Ocean Paradise	PO Box 8745		Kailua-Kona	HI	94756	US
1513	Fantastique Aquatica	Z32 999 #12A-77 A.A.		Bogota			Columbia
1551	Marmot Divers Club	872 Queen St.		Kitchener	Ontario	G3N 2E1	Canada
1560	The Depth Charge	15243 Underwater Fwy.		Marathon	FL	35003	US
1563	Blue Sports	203 12th Ave. Box 746		Giribaldi	OR	91187	US
1624	Makai SCUBA Club	PO Box 8534		Kailua-Kona	HI	94756	US
1645	Action Club	PO Box 5451-F		Sarasota	FL	32274	US

Figura 1 - Un archivio può essere visto come una tabella: le righe rappresentano i records, le colonne rappresentano i campi.



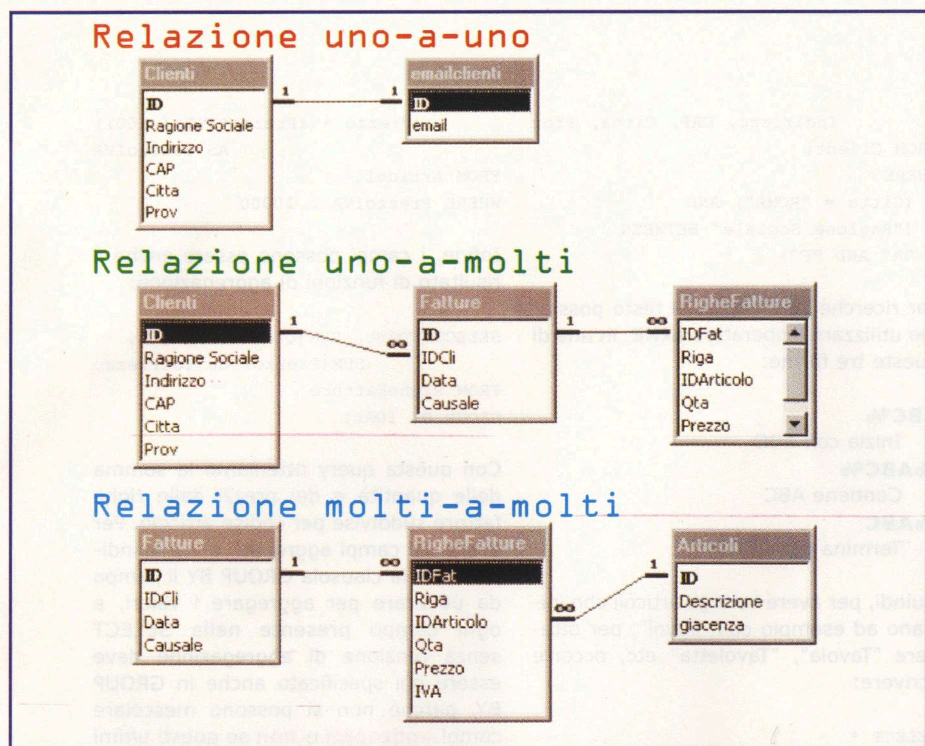


Figura 2: Tipi di relazioni possibili.

Il secondo tipo di relazione è il più comunemente utilizzato, e lega un record a molti altri, quindi solo il primo campo è **univoco**, mentre il secondo no: ad esempio per ogni cliente ci sono più fatture, ma per ogni fattura c'è un cliente solo. Questa relazione generalmente lega la chiave primaria di una tabella ad un campo della seconda tabella chiamato chiave esterna: nella tabella delle righe fatture, la chiave esterna è il numero fattura.
Infine la relazione molti a molti lega (ovviamente!) molti record a molti altri: può essere il caso delle fatture e degli articoli: con ogni fattura vengono venduti molti articoli, e ogni articolo figura in molte fatture. Nella realtà questo tipo di relazione non esiste "**puro**", ma può essere realizzato solo definendo una **terza** tabella detta di collegamento, con la quale le due tabelle da collegare hanno una relazione di tipo uno a molti. Nell'esempio di poco sopra, la funzione di collegamento viene svolta dalla tabella delle righe fatture.

SQL

Dovunque siano i vostri dati, sia che si trovino in files su disco, come i .DBF, o che siano affidati ad un server di database, raramente vi accederete direttamente; spesso vi affiderete ad un motore di accesso ai dati. Quest'ultimo è un programma a cui voi chiedete i dati e che si occupa di procurarveli: uno stadio intermedio tra il vostro programma e i database. I motori possono essere locali, ovvero sono utilizzabili solo dal computer su cui risiedono, come ad esempio il Jet di Microsoft o il BDE (Borland Database Engine) di Borland/Inprise; ed in genere accedono a tabelle residenti su files; oppure possono essere motori client/server in rete, quindi interrogabili da tutte le macchine connesse alla rete; in questo caso la residenza fisica dei dati è ignota all'utente: possono essere files come interi dischi: alcuni esempi sono PostgreSQL, Oracle, SQL Server, Interbase. In ogni caso, quasi tutti i motori di accesso ai dati sono dotati di un linguaggio di gestione dei dati, che è quasi sempre basato su SQL.
SQL (Structured Query Language) è un linguaggio definito nel 1986 e rivisto nel 1992, col nome di SQL92, divenuto lo **standard** per l'accesso ai dati e sul quale si basano tutti i motori, con l'aggiunta di varie **estensioni** specifiche di ogni produttore per gestire le **caratteristiche** peculiari di ciascun programma.
Le due componenti principali di SQL sono il Data Manipulation Language (DML) e il Data Definition Language (DDL). La prima raggruppa i comandi per la gestione dei dati, quindi la selezione, inserimento, modifica o cancellazione dei records nelle tabelle; la seconda include i comandi per la definizione dei dati, quindi la gestione di campi, tabelle, indici, chiavi etc. Le istruzioni inviate al database con SQL si definiscono query.

DML

Il Data Manipulation Language comprende 4 comandi principali:

SELECT
Seleziona i records
UPDATE
Aggiorna i records già inseriti
INSERT
Aggiunge nuovi records
DELETE
Elimina i records

Iniziamo ovviamente dalla SELECT, che è il comando utilizziamo per interrogare i database.
La sintassi completa è la seguente (si può anche mettere in un riquadro esterno):

```
SELECT [DISTINCT] * (lista_colonne)
FROM riferimento_tabelle
[WHERE espressioni]
[ORDER BY lista_ordinamento]
[GROUP BY lista_raggrruppamento]
[HAVING espressioni]
```

Il modo più semplice di utilizzarla è di chiedere tutti i campi e tutti i records da una delle nostre tabelle, di cui potete leggere la lista dei campi in una delle illustrazioni sopra:

```
SELECT *
FROM Clienti
```

Se ci servono soltanto alcuni campi, si specificano esplicitamente:

```
SELECT ID, "Ragione Sociale"
FROM Clienti
```

Le virgolette (o altro, secondo il motore utilizzato) vanno utilizzate quando il nome del campo contiene uno spazio.



Lezioni di SQL

Se vogliamo ottenere soltanto righe uniche, si utilizza la clausola DISTINCT. Ad esempio con la query

```
SELECT DISTINCT IDCli
FROM Fatture
```

otteniamo l'elenco degli identificativi dei clienti a cui abbiamo mandato almeno una fattura. Ovviamente se utilizziamo DISTINCT con un campo chiave primario o univoco, non cambia nulla.
Spesso si presenta la necessità di filtrare i dati, quindi di ottenere solo una parte dei records, che soddisfino certe condizioni: a tale scopo si utilizza la clausola WHERE; ad esempio con questa query si ottiene un elenco dei clienti col nome dalla F in poi:

```
SELECT ID, "Ragione Sociale",
              Indirizzo, CAP, Citta, Prov
FROM Clienti
WHERE "Ragione Sociale" >=    "F"
```

Gli operatori di confronto che abbiamo a disposizione sono:

> Maggiore di
>= Maggiore o uguale a
= Uguale a
<> Diverso da
<= Minore o uguale a
< Minore di

Più alcuni altri speciali operatori di confronto:

BETWEEN
Valore compreso in un intervallo
LIKE
Stringa somigliante a
IN
Valore compreso in
ANY
(usato con un operatore di confronto) Valore che soddisfa la condizione almeno con un elemento della lista
ALL
(usato con un operatore di confronto) Valore che soddisfa la condizione con tutti gli elementi della lista

Per ottenere tutti i clienti che abitano a Roma e che sono nome tra A e F si può scrivere:

```
SELECT ID, "Ragione Sociale",
     Indirizzo, CAP, Citta, Prov
FROM Clienti
WHERE
  (Citta = "ROMA") AND
  ("Ragione Sociale" BETWEEN
  "A" AND "F")
```

Per ricerche su stringhe di testo possiamo utilizzare l' operatore **LIKE**, in una di queste tre forme:

ABC%
Inizia con ABC
%ABC%
Contiene ABC
%ABC
Termina con ABC

Quindi, per avere tutti gli articoli che iniziano ad esempio con "Tavol", per ottenere "Tavola", "Tavoletta" etc, occorre scrivere:

```
SELECT *
FROM Articoli
WHERE Descrizione LIKE "TAVOL%"
```

Gli operatori IN, ANY e ALL operano invece con liste di valori: per avere ad esempio le fatture solo dei clienti con identificativo 100, 110 e 134 si scrive:

```
SELECT ID, IDCli, data
FROM Fatture
WHERE IDCli IN [100, 110, 134]
```

ANY ed ALL si utilizzano abbinati ad un altro operatore di confronto: per avere le righe fattura in cui la quantità venduta sia maggiore ad almeno uno dei valori indicati si chiede:

```
SELECT IDFat, Riga, IDArticolo, Qta
FROM RigheFatture
WHERE Qta > ANY [1000, 2500, 4200]
```

Con ALL si otterrebbero invece solo i valori maggiori di tutti i valori nella lista.
Con ognuno di questi operatori si può utilizzare la clausola NOT per invertirne l' effetto:
NOT IN [....] restituisce tutti i valori non compresi nella lista.
Come campi e valori di confronto possono essere utilizzate anche espressioni. Ad esempio per avere gli articoli il cui prezzo con IVA supera le 10.000 lire:

```
SELECT ID, Descrizione, Prezzo,
     (Prezzo + (Prezzo*IVA / 100))
                          AS PrezzoIVA
FROM Articoli
WHERE PrezzoIVA > 10000
```

Infine, i campi possono essere anche il risultato di funzioni di aggregazione:

```
SELECT IDArt, SUM(Qta) AS TotQta,
          SUM(Prezzo) as TotPrezzo
FROM RigheFatture
GROUP BY IDArt
```

Con questa query otteniamo la somma delle quantità e dei prezzi dalle righe fatture suddivise per codice articolo. Per avere dei campi aggregati, occorre indicare con la clausola GROUP BY il campo da utilizzare per aggregare i valori, e ogni campo presente nella SELECT senza funzione di aggregazione deve essere poi specificato anche in GROUP BY, perché non si possono mescolare campi **aggregati** e **non** se questi ultimi non sono usati per l' aggregazione.
Se non si usano campi di aggregazione, non occorre neppure la clausola GROUP BY; ad esempio, la query sottostante restituisce il conteggio dei clienti e il conteggio dei numeri di fax e gli indirizzi di e-mail dei clienti stessi, per sapere quanti hanno il fax e la posta elettronica:

```
SELECT COUNT(*) as TotCli,
         COUNT(email) as TotEmail,
           COUNT(Fax) as TotFax
FROM Clienti
```

Avrete intuito che COUNT(*) riporta il numero di righe inserite nella tabella.

Funzioni di aggregazione

SUM Somma
AVG Media
COUNT Conteggio
MIN Minimo
MAX Massimo

Sul prossimo numero...

Per questa volta terminiamo qua. La prossima volta torneremo sulla SELECT per vedere le **query** su più tabelle.





Jilani Khaldi - j.khaldi@oltrelinux.com

NetQuiz

soluzione client/server Windows/Linux Delphi/Interbase

Seconda Puntata

I quiz hanno sempre avuto un ruolo importante nella nostra vita lavorativa e sociale. Basta pensare agli esami di patente e di selezione, per non parlare delle varie trasmissioni televisive che hanno portato molta fortuna a tanti presentatori.

Forse questo è dovuto alla natura umana: porsi sempre più domande e cercare, ovviamente, delle risposte.

L'introduzione, quest'anno, della prova a quiz **all'esame di maturità**, è solo l'ennesima conferma della validità di tale strumento nella valutazione del grado di preparazione di un candidato.

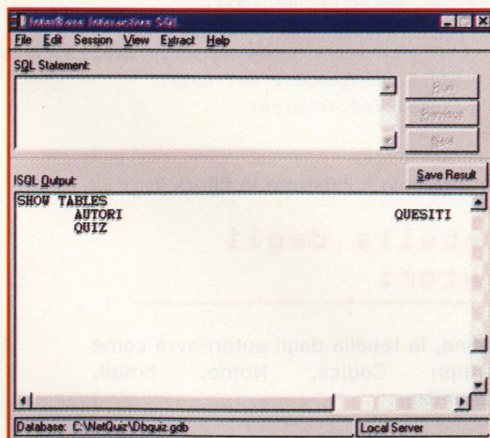
La nostra applicazione C/S (Client/Server) NetQuiz ci aiuterà a creare e svolgere una grande varietà di quiz utili per la didattica e il divertimento.

Piattaforma e strumenti di sviluppo

Come è stato accennato nella precedente puntata, NetQuiz è destinato alle scuole dotate di una Intranet con server Linux e client Windows 95/98. Lo strumento di sviluppo sarà Delphi della Borland ed Interbase come SQL server. La scelta di Linux come server è giustificata dal costo, dalla qualità del sistema operativo e dal fatto che Interbase versione 4.0 è dato in omaggio a tutti gli utenti Linux.

Lo sviluppo dell'intero programma avverrà su singolo PC usando Local Interbase (in dotazione con Delphi versione Professional e C/S) e Delphi per Windows 95/98/NT. Alla fine, quando il tutto funzionerà sul nostro singolo PC, basterà solo **copiare** il nostro database su Linux e **configurare** il server ed Interbase per avere la nostra applicazione pronta per l'uso.

Quindi, ci potrebbe seguire per lo sviluppo chiunque abbia un PC, Delphi versione Professional oppure C/S e Linux. Gli utenti invece, docenti e studenti, ci posso fornire qualche suggerimento, visto che saranno i destinatari dell'applicazione.



Definizione del database (lato server)

NetQuiz sarà soprattutto semplice da usare e da capire, a partire dalla definizione dei dati.

Considerando che ogni quiz è composto da più quesiti, realizzato da un autore, ci appare evidente che occorrono inizialmente almeno le seguenti tre tabelle: tabella dei **quiz**, tabella dei **quesiti** e tabella degli **autori**.

Per creare il nostro database, ci serviamo del programma "Windows Interbase ISQL" fornito con Local Interbase (vedi immagine 1) e del seguente codice SQL:

```
create database "dbquiz.gdb" user
"SYSDBA" password "masterkey";
```

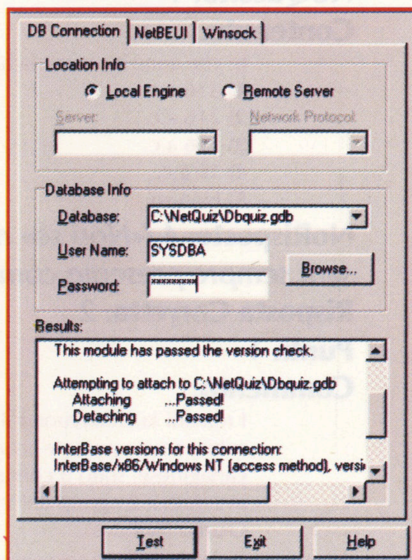
e dei seguenti domini (dati tipizzati per rendere il codice più leggibile):

```
create domain CodiceQuiz as integer
check ((value > 10000)and (value <=
99999));
```

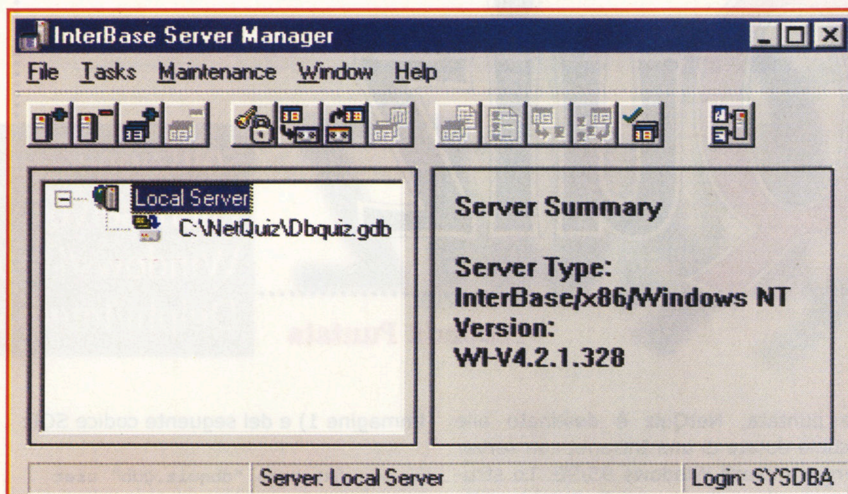
In questo caso chiediamo che ogni codice assegnato ad un nuovo quiz, sia maggiore di 10000, in modo da avere tutti i codici composti da 5 cifre.

```
create domain CodiceQuesito as
integer check ((value > 1000000)
and (value <= 9999999));
```

Lo stesso vale per il codice del quesito che sarà composto da 7 cifre, ottenuto







dalla combinazione del codice del quiz e del numero del quesito.

```
create domain CodiceAutore as
integer check ((value > 1000) and
(value <= 9999));
```

Il codice dell'autore sarà composto da 4 cifre.

```
create domain NumeroRisposte as
integer check (value in(3,4));
```

Fissa il numero delle risposte a 3 oppure a 4.

CodiceQuiz: 10001
NoQuesito: 1
Contenuto:
In che anno avvenne la battaglia di Canne?
1) 316 a.C.
2) 216 a.C.
3) 116 a.C.
4) 16 a.C.
5) Non so
NoRisposte: 4 *(«Non so» non viene conteggiata, sarà sempre presente come risposta)*
Risposta Corretta: 2
Punti: 1
Commento:
I romani la chiamarono il disastro di Canne, perché subironouna pesante sconfitta da parte dell'esercito cartaginese, sotto il comando di Annibale. La battaglia avvenne il 2 agosto del 216 a.C. sulle rive del fiume Ofanto (Aufidus).

Tabella dei quiz

La tabella dei quiz sarà composta dai seguenti campi: codice, argomento, descrizione, numero dei quesiti, autore, codice dell'autore.

Ecco un piccolo esempio:

Codice: 10001
Argomento: Storia antica
Descrizione: Storia di Roma in età repubblicana
Numero quesiti: 20
Autore: Giulio Cesare
Codice autore: 1001

Le istruzioni SQL per generare la tabella dei quiz è il seguente:

```
create table Quiz (Codice
CodiceQuiz not null,
  Argomento varchar(24) not null,
  Descrizione varchar(64) not null,
  NoQuesiti smallint not null,
  Autore varchar(24) not null,
  CodAutore CodiceAutore not null,
primary key (Codice));
```

Notate come il linguaggio SQL, essendo di tipo dichiarativo e non procedurale, riesce a spiegare quello che fa, senza la necessità di aggiungere altri commenti.

Tabella dei quesiti

Ta tabella dei quesiti avrà i seguenti campi: codice del quiz corrispondente, il numero del quesito in questione, il contenuto di quesito, il numero delle risposte , il numero della risposta corretta, i punti da dare per ogni risposta giusta e un commento (la soluzione con qualche dettaglio).
Le istruzioni SQL per generare la tabella dei quesiti sono:

```
create table Quesiti (Codice
            CodiceQuesito not null,
  NoQuesito smallint not null,
  Contenuto blob sub_type text
                segment size 1200,
  NoRisposte NumeroRisposte,
  RispostaCorretta smallint
                          not null,
  Punti smallint not null,
  Commento blob sub_type text
                 segment size 240,
  CodQuiz CodiceQuiz not null,
  primary key (Codice)
);
```

Un esempio è illustrato in figura 2.

Tabella degli autori

Infine, la tabella degli autori avrà come campi: Codice, Nome, Email, HomePage.



Esempio:

Codice: 1001
Nome: Giulio Cesare
Email: gc@SPQR.it
HomePage: Under construction!

Codice SQL per la generazione della tabella:

```
create table Autori (Codice
              CodiceAutore not null,
  Nome varchar(24) not null,
  Email varchar(40) not null,
  HomePage varchar(80),
primary key (Codice));
```

Piccoli problemi e soluzioni

Supponiamo ora che un utente si connetta al database e crei nuovi quesiti inserendo il codice di un quiz inesistente. In questo modo, avremo una serie di quesiti validi, che non fanno parte di nessun quiz del database, e quindi inutilizzabili.
La soluzione sarebbe quella di permettere la creazione di un nuovo quesito, solo se il corrispondente quiz già esiste. Questa utile funzionalità è chiamata **integrità referenziata** dei dati.
Con l'integrità referenziata, non si può cancellare un quiz che contenga quesiti. Bisogna prima cancellare tutti i quesiti che lo compongono.
Lo stesso discorso vale anche per gli autori. Se un autore non risulta che abbia realizzato almeno un quiz, non si può fare riferimento ad esso.

Il seguente codice risolve il problema sopra indicato.

```
alter table Quesiti add foreign
            key(CodQuiz) references
                     Quiz(Codice);
alter table Quiz add foreign
          key(CodAutore) references
                   Autori(Codice);
```

Un altro piccolo problema. Come mantenere ordinati i codici dei quiz, dei quesiti e degli autori in modo che, se per esempio il mio quiz attuale ha il codice 10005, il prossimo avrà il codice 10006 e non uno scelto a caso? La soluzione sarebbe quella di generare dei codici in modo automatico che siano in successione tra di loro senza l'intervento dell'utente. Il modo per farlo è quello di creare i cosiddetti ***campi incrementali***.
I campi incrementali possono essere creati tramite **trigger** oppure **stored procedure**.
Un trigger è una **procedura** che viene invocata nel database centrale, quando si verificano alcuni eventi.
Nel nostro caso d'esempio, seguenti trigger vanno ad incrementare i codici dei quiz, dei quesiti e degli autori ad ogni nuovo inserimento.

```
create trigger SetQuizGen for Quiz
active before insert as
begin
 new.Codice = gen_id(QuizGen,1);
end

create trigger SetQuesitoGen
                     for Quesiti
active before insert as
begin
 new.Codice = gen_id(QuesitoGen,1);
end

create trigger SetAutoreGen
                     for Autori
active before insert as
begin
 new.Codice = gen_id(AutoreGen,1);
end
```

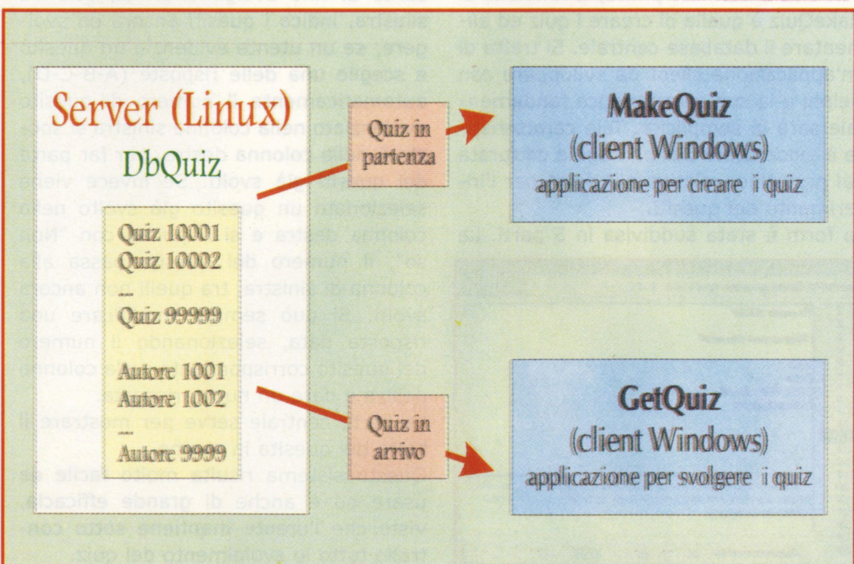
Lo stesso risultato si ottiene scrivendo **stored procedure** (*come suggerisce il nome, sono delle procedure "immagazzinate" nel database, che vengono invocate su richiesta dal programma*):

```
create procedure SetQuizGen
              returns(Num integer) as
begin
 Num = gen_id(QuizGen, 1);
end

create procedure SetQuesitoGen
              returns(Num integer) as
begin
 Num = gen_id(QuesitoGen, 1);
end

create procedure SetAutoreGen
              returns(Num integer) as
begin
 Num = gen_id(AutoreGen, 1);
end
```

Attenzione però quando si usano i trigger, Delphi non sa dell'operazione avvenuto sul server, e neppure del valore del nuovo campo "Codice" già inserito nel database. Quindi, se il nostro campo è indicizzato e serve per ordinare la nostra tabella, vediamo scomparire i dati appena inseriti. In realtà, i nostri dati non sono spariti, ma sono stati semplicemente spostati, visto che l'ordinamento dei dati avviene in modo automatico dopo ogni nuovo inserimento.
Comunemente, per i campi incrementali, si usano i trigger e non le stored procedure.





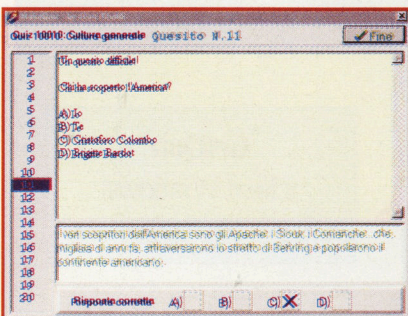
Linux&Scuola

A che punto siamo?

Finora abbiamo definito il database, i vari campi e i controlli su alcuni valori critici. A questo punto, possiamo già iniziare a creare i nostri quiz, inserendoli direttamente tramite procedure SQL, utilizzando Windows Interbase ISQL precedentemente presentato. Sarà sicuramente una soluzione valida, ma è molto complicata, insidiosa e richiede una discreta conoscenza di SQL da parte dell'utente.
La soluzione ideale sarebbe quella di creare una comoda applicazione Windows, ad interfaccia grafica, senza utilizzare direttamente codice SQL. A questa condizione, per qualsiasi utente, sarà possibile realizzare quiz, semplicemente scrivendo testi e premendo pulsanti.
Alla fine, verrà realizzata l'applicazione client, che accede al server, preleva un quiz e lo svolge.
Quindi, l'applicazione NetQuiz è composta da due programmi client (MakeQuiz e GetQuiz) che accedono ad un database sul server (DbQuiz). MakeQuiz, come indica il nome, crea nuovi quiz per alimentare il database; GetQuiz, dal nome abbastanza suggestivo, carica un quiz da server per essere svolto.

Realizzazione di MakeQuiz

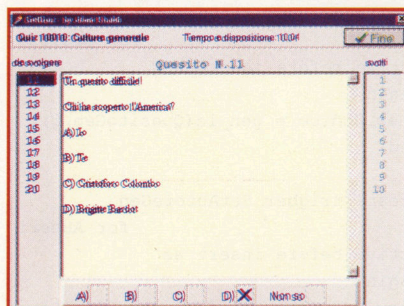
Come ho accennato prima, la funzione di MakeQuiz è quella di creare i quiz ed alimentare il database centrale. Si tratta di un'applicazione client da sviluppare con Delphi e la sua caratteristica fondamentale sarà la semplicità. Tale caratteristica è evidenziata dall'immagine catturata dal prototipo, relativa alla form per l'inserimento dei quesiti.
La form è stata suddivisa in 5 parti. La



parte in alto indica il tema del quiz e il numero del quesito. La colonna sinistra, indica il numero dei quesiti da svolgere. L'utente, dopo aver evidenziato il numero del quesito da creare, può passare alla parte centrale (gialla) e digitare il testo corrispondente, poi passare alla casella bianca sotto, ed inserire il commento relativo alla soluzione. Infine la parte più in basso, con le indicazioni A-B-C-D, serve per indicare la risposta corretta.

Realizzazione di GetQuiz

GetQuiz è l'applicazione per svolgere i quiz. Sarà maggiormente utilizzata dagli studenti. Dall'immagine si intuisce la grande somiglianza con MakeQuiz, ma



con qualche piccola differenza.
La form è stata suddivisa in 6 parti. La parte più in alto indica il tema del quiz, il tempo ancora a disposizione e il pulsante di fine svolgimento. La colonna sinistra, indica i quesiti ancora da svolgere; se un utente evidenzia un quesito e sceglie una delle risposte (A-B-C-D), automaticamente il numero di quesito evidenziato nella colonna sinistra si sposterà nella colonna destra, per far parte dei quesiti già svolti. Se invece viene selezionato un quesito già svolto nella colonna destra e si risponde con "Non so", il numero del quesito passa alla colonna di sinistra, tra quelli non ancora svolti. Si può sempre modificare una risposta data, selezionando il numero del quesito corrispondente nella colonna destra e dare un nuova risposta.
La parte centrale serve per mostrare il testo del quesito in esame.
Questo sistema risulta molto facile da usare ed è anche di grande efficacia, visto che l'utente mantiene sotto controllo tutto lo svolgimento del quiz.

Premendo il tasto "Fine", avremo l'esito della nostra interrogazione: numero delle risposte giuste, nulle e sbagliate (con il relativo commento), il punteggio e il tempo impiegato per lo svolgimento.
Aspetto i vostri consigli e suggerimenti. Mi rivolgo in modo particolare ai docenti e agli studenti, destinatari di NetQuiz.



```
/* Crea un database per i dbquiz */
create database "dbquiz.gdb" user "SYSDBA" password "masterkey";

/* Definizione domini */
create domain CodiceQuiz as integer check ((value > 10000) and
(value <= 999999));
create domain CodiceQuesito as integer check ((value > 1000000)
and (value <= 9999999));
create domain NumeroRisposte as integer check (value in(3,4));
create domain CodiceAutore as integer check ((value > 1000) and
(value <= 9999));

/* Tabella: Quiz, Proprietario: SYSDBA */
create table Quiz (Codice CodiceQuiz not null,
        Argomento varchar(24) not null,
        Descrizione varchar(64) not null,
        NoQuesiti smallint not null,
        Autore varchar(24) not null,
        CodAutore CodiceAutore not null,
        primary key (Codice)
);

/* Tabella: Quesiti, Proprietario: SYSDBA */
create table Quesiti (Codice CodiceQuesito not null,
        NoQuesito smallint not null,
        Contenuto blob sub_type text segment size 1200,
        RispostaCorretta smallint not null,
        Punti smallint not null,
        Commento blob sub_type text segment size 240,
        CodQuiz CodiceQuiz not null,
        primary key (Codice)
);

/* Tabella: Autori, Proprietario: SYSDBA */
create table Autori (Codice CodiceAutore not null,
        Nome varchar(24) not null,
        Email varchar(40) not null,
        HomePage varchar(80),
primary key (Codice));

/*  Definizione degli indici per le tabelle */
create index Idx_Argomento on Quiz(Argomento);
create index Idx_Autore on Quiz(Autore);
alter table Quesiti add foreign key(CodQuiz) references
Quiz(Codice);
alter table Quiz add foreign key(CodAutore) references
Autori(Codice);

create generator QuizGen;
set generator QuizGen to 10000;
create generator QuesitoGen;
set generator QuesitoGen to 1000000;
create generator AutoreGen;
set generator AutoreGen to 1000;

/* Triggers */
set term ! ;
create trigger SetQuizGen for Quiz
active before insert position 0 as
begin
  new.Codice = gen_id(QuizGen, 1);
end !

create trigger SetQuesitoGen for Quesiti
active before insert position 0 as
begin
  new.Codice = gen_id(QuesitoGen, 1);
end !

create trigger SetQuizAutore for Autori
active before insert position 0 as
begin
  new.Codice = gen_id(AutoreGen, 1);
end !
commit work !
set term ; !
```

FINALMENTE LIBERI DI SCRIVERE TeX

RICCARDO MURRI
r.murri@oltrelinux.com

Laureando in Matematica, si interessa al mondo Linux da diversi anni. Oltre al TeX, è appassionato di HTML.

Dopo aver presentato il TeX lo scorso numero, iniziamo ad analizzare i primi documenti e le prime parole chiave di questo linguaggio. Dopo questa puntata sarete già in grado di scrivere del testo formattato.

Prima di cominciare, raduniamo gli ingredienti che occorrono per scrivere con LATEX. Se già avete un sistema TEX funzionante (per esempio, usate i computer dell'Università e qualcuno l'ha già installato e configurato), saltate oltre: il TEX è ampiamente portabile, e quasi nulla di quello che dirò è specifico o ristretto ad una particolare architettura (avrete gli stessi risultati con teTEX per FreeBSD o con MiKTEX per MS-Windows).

Editor di Testi

Ogni editor di testi va bene per scrivere in LATEX; due in particolare però offrono supporto specifico per i file TEX: **LyX** ed **EMACS** (tramite il modo AUC-TEX). Ognuno ha comunque il suo editor preferito: non è il caso di cambiare ora, TEX non fa differenze.

TEX

Servirà ovviamente il **compilatore** TEX, un programma per vedere i file **.dvi** (xdvi, kdvi), ed uno per eventualmente **stamparli** (dvips, dvilj). TeTEX è una distribuzione di TEX per i sistemi UNIX, comprendente questi tre ed altri programmi complementari, i più diffusi pacchetti di macro come LATEX e Plain-TEX, ed infine un'abbondante documentazione.
Se non avete niente di tutto questo, ecco una rapida lista di cose da installare per avere teTEX pronto:

Caldera
installate il pacchetto TeTeX.

Debian
installate i pacchetti tetex-base, tetex-bin, tetex-doc e tetex-extra;

FreeBSD, OpenBSD
teTEX è fra i programmi che si possono installare con il sistema dei ports: usate quindi il relativo Makefile sotto la directory /usr/ports;

NetBSD
Esiste una versione di teTEX preparata per il sistema di pacchetti di NetBSD: installate teTeX, teTeX-bin, teTeX-share, teTeX-sharesrc.

Red Hat
come minimo sono necessari i pacchetti tetex, tetex-latex, tetex-doc, tetex-xdvi. Se volete stampare, installate anche tetex-dvips (stampanti POSTSCRIPT) o tetex-dvilj (HP LaserJet e compatibili PCL);

Slackware
installate i tre pacchetti tetex.tgz, tex_bin.tgz e tex_doc.tgz, dalla serie tex1;

SuSE
installate i pacchetti tetex, te_ams, te_cont, te_dvilj (solo per stampare su stampanti HP LaserJet o compatibili), te_kpath, te_latex.

MS-Windows
Se proprio non potete riformattare l'hard-disk per metterci Linux, troverete tutte le istruzioni per reperire ed installare MiKTEX a **http://www.esat.kuleuven.ac.be/minten/NTTeXing/NTTeXing.html**

Chi usa qualche altro sistema, come pure chi vuole compilare da sé tutti i programmi che installa, può trovare le sorgenti di teTEX all'URL: **ftp://ftp.dante.de/tex-archive/systems/unix/teTeX/.**
Chi usa EMACS e volesse provare AUC-TEX, troverà tutte le informazioni necessarie a **http://sunsite.auc.dk/auctex/**; chi usa Debian o Red Hat può anche subito installare il pacchetto relativo, ma le altre distribuzioni non sembrano averlo disponibile.

Qui comincia l'avventura...

Ricordiamo brevemente il funzionamento del sistema TEX: il programma tex **prende in ingresso un file .tex** e lo **compila in un file .dvi** (questo è un particolare formato che memorizza tutte le informazioni necessarie per la resa tipografica di un documento, sia che questa sia fatta su una stampante o su un video). Il file .dvi viene poi **convertito in un file POSTSCRIPT** od un altro formato pronto per la stampa, per mezzo di appositi convertitori. La prima fase prende il nome di compilazione del file .tex: qui è necessario fare molta attenzione, perché un errore di

sintassi nei comandi LATEX è spesso sufficiente per arrestare il procedimento o comunque per creare risultati inaspettati da quel punto in avanti. La seconda fase è **completamente automatica** e non avrete mai problemi se il software è installato correttamente, anche se può richiedere molto più tempo, specie sui computer lenti.
Per compilare un file .tex si usa il comando latex: ad esempio per compilare il file gino.tex si dovrà scrivere

```
latex gino.tex
```

che crea il file gino.dvi.
Per avere una anteprima del file gino.dvi, si usa il comando xdvi sotto X/Windows: ad esempio,

```
xdvi gino.dvi
```

La struttura di un file LATEX

Un file LATEX è un comune file di testo, dove vengono inserite opportune **indicazioni** (**comandi**) per controllare la composizione **tipografica** finale. Tutti questi comandi sono scritti con i normali caratteri ASCII, quelli di una tastiera americana, per intenderci.
Un file LATEX è diviso in due parti principali: il **preambolo** ed il **corpo del documento**. Il corpo del documento contiene tutto il testo che viene reso effettivamente in stampa, ed è racchiuso tra i comandi `\begin{document}` e `\end{document}`.
Il documento termina con `\end{document}`: nulla verrà più letto da TEXoltre questa riga.
Il testo è scritto normalmente, con i paragrafi separati da una o più righe vuote; LATEX **ignora** ogni **fine-riga** che non sia seguita da una riga vuota, ed accumula tutto il testo in un **capoverso**. Pure, non fa differenza la quantità di spazi che inserite tra una parola e l'altra: LATEX regola da sé la distanza tra le diverse parole, ed allinea il testo sui margini della pagina. In questo modo, potete inserire tabulazioni e spazi nel vostro file .tex per renderlo più leggibile (come si fa con il codice sorgente dei programmi) mentre lo state modificando, certi comunque che il **risultato** finale **non cambierà**. Il preambolo è costituito da tutto il materiale che si trova prima del comando `\begin{document}`: per definizione il preambolo non può contenere testo da stampare, ma solo comandi LATEX per **impostare** la **struttura globale** del documento, ad esempio per cambiare le dimensioni della pagina, spostare i margini, o caricare estensioni LATEX per usi particolari. (*Per chi conosce HTML, preambolo e corpo di un documento in LATEX corrispondono più o meno agli elementi* HEAD *e* BODY)
Esaminiamo sommariamente il listato di figura 1.

```
\documentclass[a4paper]{article}

% Le due righe seguenti caricano estensioni di LaTeX
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}

% qui finisce il preambolo LaTeX
\begin{document}

\section{Premessa}

Le licenze per la maggioranza dei programmi hanno lo scopo di togliere
all'utente la libertà di condividerlo e di modificarlo. Al contrario,
la Licenza Pubblica Generica GNU è intesa a garantire la libertà di
condividere e modificare il free software, al fine di assicurare che i
programmi siano ``liberi'' per tutti i loro utenti.  Questa Licenza si
applica alla maggioranza dei programmi della Free Software Foundation
e ad ogni altro programma i cui autori hanno scelto questa Licenza.
Alcuni altri programmi della Free Software Foundation sono invece
coperti dalla Licenza Pubblica Generica per Librerie.  Chiunque può
usare questa Licenza per i propri programmi.

Quando si  parla di ``free  software'', ci si riferisce  alla libertà,
non al prezzo.  Le nostre  Licenze (la GPL  e la LGPL) sono progettate
per assicurarsi che ciascuno abbia la libertà di distribuire copie del
free   software (e farsi pagare  per   questo, se  vuole), che ciascuno
riceva il codice sorgente o che lo  possa ottenere se lo desidera, che
ciascuno possa  modificare il programma o usarne  delle parti in nuovi
programmi ``liberi'' e che ciascuno sappia di potere fare queste cose.

[...]

\end{document}
```

Figura 1: l'inizio della licenza GNU

I comandi LATEX **iniziano** tutti col simbolo \, i brani da stampare sono scritti liberamente, come in un normale file di testo. Ogni riga che comincia con **%** è un commento2 e come tale viene ignorato da LATEX. Il preambolo LATEX contiene tre comandi: `\documentclass` e `\usepackage` (usato due volte). `\documentclass` deve essere sempre dichiarato nella prima riga utile del file, perché specifica quale formato LATEX dovrà utilizzare per l'impaginazione del documento. La prima invocazione di `\usepackage` comunica a LATEX che stiamo scrivendo in lingua italiana (in modo che scriva, ad esempio, "9 Settembre 1999" e non "September 9, 1999"). La seconda chiamata a `\usepackage` permette a LATEX di **interpretare correttamente** le **lettere accentate**. Tralasciamo per ora i dettagli su questi comandi; raccomando solo di usare queste due invocazioni di `\usepackage` in ogni documento che scriverete in Italiano -- per l'Inglese sono entrambe inutili.
Il corpo del documento contiene due paragrafi di testo: notate come il primo paragrafo sia stato scritto senza troppo riguardo per i margini, mentre il secondo è perfettamente allineato per una ampiezza di 72 caratteri; eppure entrambi verranno da LATEX giustificati esattamente (vedi figura 2. Il comando `\section{Premessa}` inizia una sezione intitolata appunto Premessa.

\documentclass	[a4paper]	{article}
comando	opzioni	argomento

Il primo comando del nostro file...



Anatomia di un comando

Vediamo da vicino com'è fatto un comando LATEX, prendendo ad esempio la prima riga del nostro file.
Ogni **comando** comincia col carattere '\', seguito da una o più lettere. **Non** sono ammessi caratteri **non letterali** nel nome di un comando: `\interlinea_2` non è un nome ben formato, mentre `\InterlineaDoppia` sì. Fanno eccezione i nomi di un solo simbolo, che allora può essere un qualunque carattere valido. LATEX **distingue** tra **maiuscole** e **minuscole** nei nomi di comandi: `\large` e `\Large` sono comandi diversi, con effetti diversi.

L'argomento di un comando è sempre del testo racchiuso dalle parentesi graffe. Cosa accade a questo testo dipende, ovviamente, dal comando:

```
\documentclass{article}
```

comincia un documento di classe article (articolo), mentre

```
\section{L'influenza del pane
imburrato sulla rotazione
terrestre}
```

comincia una sezione intitolata, ovviamente, «*L'influenza del pane imburrato sulla rotazione terrestre*». Alcuni comandi accettano **più di un argomento**, ciascuno racchiuso dalle sue parentesi: ad esempio,

```
\frac{x}{y}
```

stampa una frazione con x al numeratore e y al denominatore. LATEX ignora eventuali spazi tra il comando ed il suo argomento, come pure tra gli argomenti, se ce n'è più d'uno; tutte le seguenti forme sono **equivalenti**:

```
\frac{x}{y}
\frac {x}{y}
\frac {x} {y}
```

Quando un argomento è costituito da un solo carattere, si possono omettere le parentesi graffe. **Attenzione**! Questa sintassi non dà modo a TEX di controllare che il numero di argomenti passati ad un comando sia quello giusto! Ricordate che ogni comando accetta solo un numero prefissato di argomenti, ma se ne sbagliate il numero non avrete nessun **errore sintattico**; se ne mettete **di più**, quelli residui verranno considerati come normale testo; se ne fornite **di meno**, quelli che mancano verrano presi dal testo seguente (questo può generare effetti strani).
Le opzioni sono sempre racchiuse fra parentesi quadre, e sono, come dice il nome, facoltative: selezionano varianti nel comportamento di un comando. Per esempio,

```
\documentclass[a4paper]{article}
```

impagina un articolo per la stampa su carta di formato A4. I comandi che accettano un solo argomento a volte accettano anche più di una opzione: in questo caso le separiamo con una virgola; per esempio,

```
\documentclass[a4paper,twocolumn]{article}
```

impagina l'articolo in formato A4 e su due colonne. I comandi che accettano più argomenti, invece, vogliono le diverse opzioni interposte tra un argomento e l'altro. Le opzioni sono anche chiamate argomenti opzionali, perché il loro comportamento è in effetti del tutto simile a quello degli argomenti.

Titoli ed autori

Cominciamo a lavorare un po' sul documento di figura 1: come si fa ad aggiungere un titolo, il nome dell'autore e la data? Un'intestazione insomma? LATEX divide questo compito in due parti:
1. dichiarazione di titolo, autore e data con i comandi `\title`, `\author` e `\date`;
2. creazione della pagina del titolo con `\maketitle`.

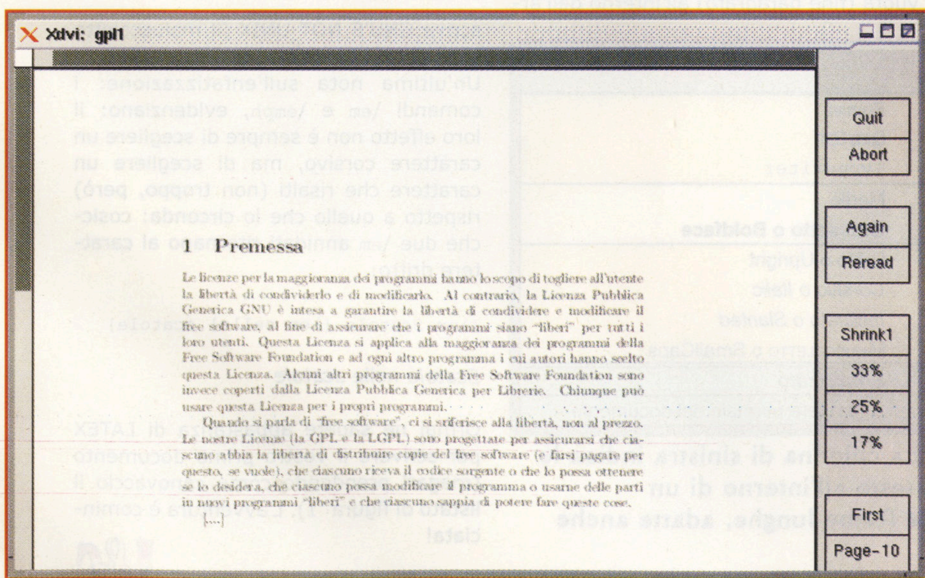
Ovviamente le dichiarazioni devono precedere la chiamata \maketitle. Tutti questi comandi vanno inseriti nel corpo del documento.
Il comando `\title` accetta un solo argomento: il testo che comporrà il titolo del documento; per esempio,

```
\title{Licenza Pubblica Generica
GNU}
```

Eventuali sottotitoli possono essere aggiunti separandoli dal primo titolo con \\:

```
\title{Licenza Pubblica Generica
GNU \\ Versione 2}
```

Anche il comando \author vuole un solo argomento, il nome dell'autore. Oltre al nome, si possono indicare altre informazioni come un recapito postale o l'indirizzo e-mail, basta separare le righe con \\:



1 Premessa

Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all'utente la libertà di condividerlo e di modificarlo. Al contrario, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano "liberi" per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e ad ogni altro programma i cui autori hanno scelto questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica per Librerie. Chiunque può usare questa Licenza per i propri programmi.

Quando si parla di "free software", ci si sriferisce alla libertà, non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del free software (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi "liberi" e che ciascuno sappia di potere fare queste cose.

[...]

Figura 2: Ecco come LaTex trasforma il listato di figura 1



Corso di TeX

```
\author{Free Software Foundation,
Inc. \\
  675 Mass Ave, \\
  Cambridge, MA 02139, \\
  USA}
```

Si possono anche specificare più autori, basta separare le relative informazioni con \and:

```
\author{Free Software Foundation,
Inc. \\
  675 Mass Ave, \\
  Cambridge, MA 02139, \\
  USA \\
  (originale inglese)
  \and
  Gruppo PLUTO e ILS \\
  Casella Postale 400 \\
  17000 Savona \\
  Italia \\
  (traduzione italiana)}
```

Infine, \date specifica la data in cui il documento è stato composto; in mancanza di indicazione, LATEX inserirà la data in cui il file viene compilato. All'interno dell'argomento di \date, potete usare il comando \today per indicare la data odierna.

Il comando \maketitle non prende argomenti e può essere posizionato dove volete all'interno del corpo del documento: \maketitle apre una nuova pagina per comporre l'intestazione, tutto quello che viene prima apparirà nelle facciate precedenti il titolo.

Per prova, inserite \title, \author, \date e \maketitle nel listato di figura 1, in modo da produrre un titolo come in figura 3.

Evidenziare il testo

Il comando per **evidenziare** una porzione di testo è \emph, che accetta un solo argomento: il testo da evidenziare. Ad esempio,

```
\emph{Licenza Pubblica Generica
GNU} vs. Licenza Pubblica Generica
GNU
```

produce

Licenza Pubblica Generica GNU vs. Licenza Pubblica Generica GNU

Esiste anche un'altra **forma di evidenziazione**, che produce **uguali effetti tipografici**, ma ha una sintassi differente:

```
{\em Licenza Pubblica Generica GNU}
```

La **differenza** sta in questo: con la seconda forma si possono evidenziare interi brani di testo, anche comprendenti più paragrafi, mentre la prima è limitata ad una porzione di paragrafo: LATEX darà errore se incontra una riga vuota (fine paragrafo) all'interno dell'argomento di \emph. Per prova, evidenziate Licenza Pubblica generica GNU ogni volta che compare nel listato di figura 1.

Forma Breve	Forma Lunga	Effetto
\textrm{...}	{\rmfamily ...}	Roman
\textsf{...}	{\sffamily ...}	Sans Serif
\texttt{...}	{\ttfamily ...}	Typewriter
\textmd{...}	{\mdseries ...}	Medio
\textbf{...}	{\bfseries ...}	**Grassetto o Boldface**
\textup{...}	{\upshape ...}	Dritto o Upright
\textit{...}	{\itshape ...}	*Corsivo o Italic*
\textsl{...}	{\slshape ...}	*Inclinato o Slanted*
\textsc{...}	{\scshape ...}	MAIUSCOLETTO o SmallCaps
\emph{...}	{\em ...}	***Enfatizzato***
\textnormal{...}	{\normalfont ...}	Il carattere normale del documento

I vari stili dei caratteri in LATEX. La colonna di sinistra indica le forme brevi, adatte per porzioni di testo all'interno di un capoverso; quella centrale elenca le forme lunghe, adatte anche per più paragrafi.

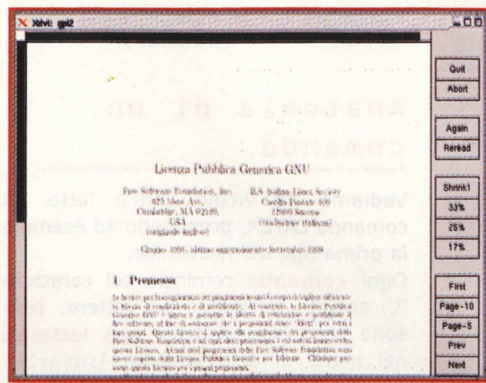


Figura 3: La licenza pubblica GNU col suo bel titolo!

LATEX permette molti altri effetti sul testo, e tutti ammettono le due forme sintattiche alternative: guardate la tabella di figura 4 per un elenco completo.

Osservate che i comandi che cambiano lo stile del carattere sono divisi in tre categorie: **family**, **series** e **shape**. I comandi appartenenti a categorie diverse combinano i loro effetti:

```
{\itshape\bfseries GNU's Not UNIX}
```

produce una scritta in corsivo grassetto, mentre di più comandi della stessa categoria, l'ultimo prevale:

```
{\itshape\scshape GNU's Not UNIX}
```

scrive GNU'S NOT UNIX (il maiuscoletto soppianta il corsivo).

Un'ultima nota sull'enfatizzazione: i comandi \em e \emph, evidenziano: il loro effetto non è sempre di scegliere un carattere corsivo, ma di scegliere un carattere che risalti (non troppo, però) rispetto a quello che lo circonda: cosicchè due \em annidati ritornano al carattere dritto:

```
{\em Scatole {\em nelle} scatole}
```

Scatole nelle *scatole*

Ormai ne sapete abbastanza di LATEX per scrivere il vostro primo documento (magari prendendo come canovaccio il listato di figura 1). L'avventura è cominciata!





INTRODUZIONE AI GCC

the gnu c compiler

editor ed ambienti integrati di programmazione

FELICE MAINOLFI

f.mainolfi@oltrelinux.com

25 anni, laureando in Ingegneria Elettronica, si occupa di sistemi distribuiti e sicurezza in ambiente unix.

Una presentazione di alcune dei più diffusi editors per programmare in C: alcuni sono editor generici, altri sono front-end grafici che si integrano perfettamente con il compilatore, che niente hanno da invidiare ai "cugini" presenti su altri sistemi, Windows in testa. Una breve introduzione che verrà ampliata al più presto.

Una delle prime cose che colpiscono di linux, nel bene e nel male, è che una parte significativa della gestione del sistema è svolta sulla **linea di comando**. Anche il modo di compilare i programmi è ancora "old style": un programma viene **editato**, **compilato** ed eventualmente **linkato**, nella maggior parte dei casi sulla linea di comando. Soprattutto per **grandi** progetti la situazione può diventare scomoda. In questo articolo spazieremo dai semplici text editor a veri e propri ambienti integrati, cercando di focalizzarne le caratteristiche salienti per dar modo di scegliere l'ambiente o l'editor che più si avvicina alle proprie esigenze.

L'editor vi

Chi di voi non ha mai utilizzato vi da quando usa Linux? Nessuno, spero. Se probabilmente ha la palma dell'editor **più usato** non è certamente un buon esempio di programma "user friendly". Soprattuto all'utente **meno esperto** il funzionamento di vi può apparire quanto meno criptico. Allora perchè usarlo? Perchè non passare direttamente ad editor meno complicati? Il motivo è semplice: vi è presente in **tutti** gli ambienti unix e certamente può capitare che sia l'unico editor **installato** nel sistema. È utile, quindi conoscerne almeno i comandi basilari, in modo da cavarcela anche nelle situazioni più disagiate. Per avviarlo è sufficiente il comando

```
vi nomefile
```

esistono diversi parametri che possono essere passati sulla linea di comando. Molto utile quando si procede al debug di un programma è la possibilità di avviare vi con il cursore già posizionato in una certa linea di codice. Per esempio

```
vi +20 programma.c
```

avvia vi posizionando il cursore alla linea 20 del file programma.c. Entrati nel programma esso si trova in **modalità comandi**: ogni tasto premuto sarà considerato un comando e non un testo da inserire. Premendo il tasto **':'** si passa alla modalità linea di comando nella quale le azioni da compiere vengono impartite inserendole nell'ultima linea dello schermo ed **eseguite** quando si preme invio. Dalla modalità comandi, premendo il tasto 'I', si passa alla modalità inserimento nella quale si può introdurre il testo in maniera tradizionale. Si può passare in modalità comando indipendentemenete dallo stato corrente premendo una o più volte il tasto ESC. Ogni modalità è caratterizzata da un certo numero di comandi. Esistono una serie di cloni di vi (vim, elvis...) che ne ampliano le potenzialità e che spesso sostituiscono vi: il più delle volte vi non è altro che un **link** ad un suo clone. Nella suse 6.2, per esempio, è un link a vim.

joe: un editor in stile wordstar

Solo chi aveva le mani su una tastiera più di 10 anni fa può ricordare il mitico Wordstar, un wordprocessor per MSDOS che a suo tempo era diffusissimo. A causa del suo successo molte delle sequenze di tasti associate ai suoi comandi furono copiate da altri programmi. Tra questi gli ambienti di sviluppo integrati

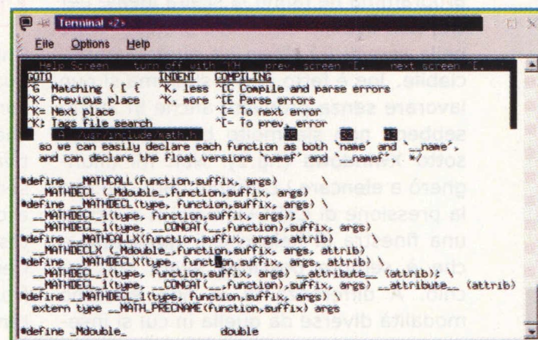


Figura 1: joe in una finestra X, non molto bello ma funzionale.



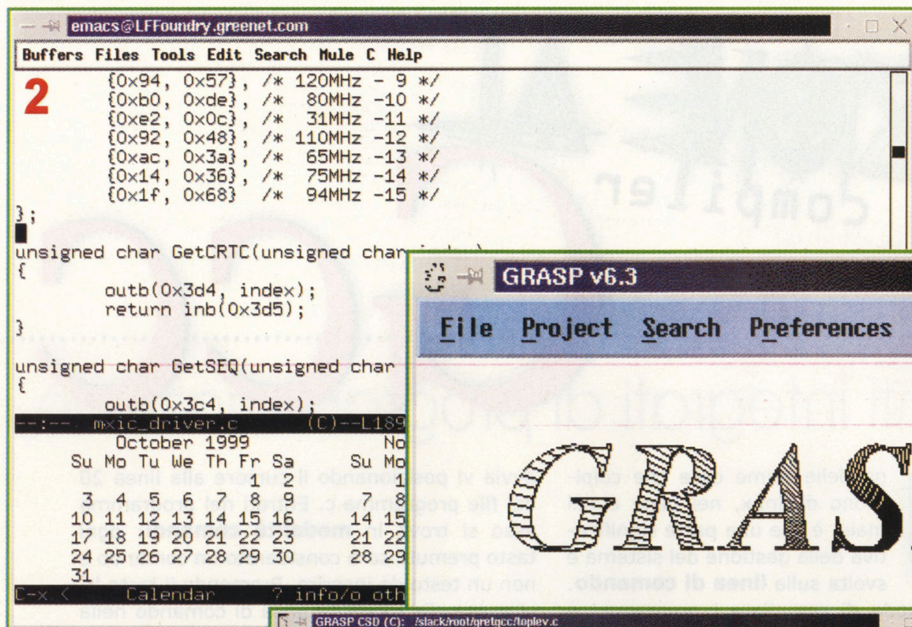




Fig. 2: emacs riconosce di essere sotto Xwindows e gestisce mouse e menu.
Fig. 3: la schermata iniziale di grasp e' anche la finestra per gestire i progetti.
Fig. 4: grasp all'opera. Notare il CSD associato al codice.

(IDE) per il Turbo C e il Turbo Pascal. Joe è un editor che ricalca quel set di comandi e ciò ne fa una scelta **appetibile** per chi ha avuto esperienza con quei compilatori. L'indentazione, la semplicità con cui è possibile gestire i blocchi, la stessa capacità di compilare un programma ne fanno la scelta ideale per chi, non avendo eccessive pretese, vede nella semplicità d'uso un punto irrinunciabile. Joe è fatto per la shell ma si può lavorare senza problemi anche in xterm sebbene non sia molto bello a vedersi sotto Xwindows (fig.1). Non mi dilungherò a elencare i comandi dato che con la pressione di CTRL+K e poi H si attiva una finestra di help che li **riassume** e che è sempre possibile avere sott'occhio. A differenza di vi non esistono modalità diverse da quella in cui si inserisce il testo e questo lo rende più facile da imparare. Non è presente in alcune distribuzioni, per cui potreste avere la necessità di scaricarloda **ftp://ftp.std.com/src/editors/joe2.8.tar.Z.**

Emacs

Chiamare emacs editor di testi sarebbe quanto meno **riduttivo**: tante e tali sono le possibilità di questo programma che qualsiasi definizione sarebbe limitativa. Sebbene al primo impatto possa sembrare **ostico** e **difficile** da imparare a causa delle innumerevoli sequenze di tasti, mano mano che si acquista esperienza ci si rende conto di trovarsi di fronte a qualcosa di molto potente. Non tentiamo neanche di descriverlo in questo piccolo spazio: non basterebbe l'intera rivista per farlo. Il suo utilizzo è comunque ben documentato dal sistema di help di cui dispone: potete attivarlo premendo **C-h** (nella convenzione emacs significa **CTRL + H**). Vi ricordo inoltre che il tasto "**META**", in emacs usato molto spesso, corrisponde al tasto **ESC**. Se avviate emacs da Xwindows il programma si configurerà in modo da interagire correttamente con il **mouse** e fornirà un comodissimo menu contestuale nella parte alta della finestra (figura 2).

Editor ed ambienti di sviluppo X

Per chi ha acquistato esperienza in ambienti di sviluppo sotto windows il dover operare con sequenze di tasti può essere disorientante. Se poi c'è la necessità di trattare grossi progetti che comportano la gestione contemporanea di più file, può essere comodo ricorrere ad un IDE grafico. Per Xwindows ne esistono diversi. Alcuni sono generici e permettono di compiere al loro interno le operazioni fondamentali di **editing**, **compilazione** e **dubugging**. Altri, invece, si rivolgono ai programmatori che fanno uso di librerie specifiche. Così, per citarne uno, troviamo il **QTarchitect**, uno strumento "visual" per costruire applicazioni che usano le QT. In questo ambito tratteremo solo gli IDE veri e propri, in quanto ci interessa esaminare quei tool rivolti al programmatore che non necessariamente deve sviluppare per X.

Grasp

Grasp (figura 3) è fondamentalmente un editor di testo per **programmatori**. La sua caratteristica fondamentale è che costruisce anche il **CSD**, una sorta di diagramma a blocchi, del programma che si sta redigendo. Inoltre l'utilizzo dei colori per evidenziare la sintassi del codice consente di riconoscere al volo



eventuali errori grossolani. In figura 4 potete osservare come siano evidenziate le varie parti del codice di una funzione: la dichiarazione, i blocchi condizionali, i punti di uscita. Questo ne fa uno strumento appetibile per chi ha necessità di mettere mano su codice prodotto da **terzi** o che debba procedere alla **manutenzione** del proprio codice. Non mancano la gestione di progetti (ovvero di codice articolato su più file) e le macro. Il supporto dei **template**, ovvero di costrutti "prefabbricati" personalizzabili rende la produzione del codice ancora più veloce. I messaggi del compilatore e l'output del programma vengono "inquadrati" in apposite finestre, rendendo il **catching** degli errori meno problematico. Il programma è disponibile, anche per windows, all'url http://www.eng.auburn.edu/department/cse/research/grasp/ Grasp è ottimizzato non solo per C e C++ ma anche per ADA95, Java e VHDL (un linguaggio per l'implementazione di circuiti digitali).

Code crusader

Questo ambiente integrato **free** è ispirato al più famoso code warrior che però è commerciale. A detta degli autori code crusader supera in molti punti il suo "cugino" non free (http://www.cco.caltech.edu/~jafl/jcc). Utilizza il framework Jx che non è proprio diffusissimo, tuttavia presenta un numero di funzionalità impressionante. Date una occhiata in figura 8; cosa vedete? ebbene si, è **HTML**! Code crusader permette l'editing oltre che dei programmi in c/c++ anche di html con tanto di possibilità di ottenere una preview. Per gli aspetti attienenti il c segnalo la possibilità di creare i file header automaticamente e sopratutto anche la produzione dei makefile mediante l'utility esterna makemake è completamente automatizzata (figura 7). Per il c++ è presente anche un **browser** per navigare nella **gerarchia** di **classi**. Insomma... una vero ambiente di sviluppo che non ha nulla da invidiare rispetto ai più blasonati ambienti per Windows.

Conclusioni

Questa breve rassegna non vuole certo esaurire l'argomento. Esistono molti altri IDE a disposizione, alcuni per l'utente alle prime armi, altri per programmatori esperti. Quello che abbiamo cercato di fare è di dare **un'idea** di quello che offre la scena



5

About

Code Crusader 2.1.4

Text editor and UNIX development enviroment

Copyright © 1996-99 by John Lindal.
This program may be freely distributed at no charge.

6

utmpview.c

Files File Edit Search Project Preferences Help Header

```
#include <stdio.h>
#include <utmp.h>
#include <fcntl.h>
#include <string.h>
#include <signal.h>
#include <sys/socket.h>
#include <netinet/in.h>

int write_data(int s, char *buf, int n)
{
    int bcount; /* contatore dei byte scritti */
    int br;     /* byte scritti in un ciclo */

        bcount = 0;
        br = 0;
        while (bcount < n) {          /* loop finche' il buffer non e' vuoto */
            if ((br = write(s, buf, n-bcount)) > 0) {
                bcount += br;
                buf += br;      /* muovi il puntatore del buffer per
                                   la prossima scrittura */
            }
        else if (br < 0)              /* segnala un errore al chiamante */
            return(-1);
        }
        return(bcount);
}
```

Line 11 /root/utmpview.c Read only

7

prova.jcc

Files File Edit Project Source Preferences Help Config

Make Run

New group
alloca.c
acconfig.h
aclocal.m4
assert.h
auto-config.h
basic-block.h
bc-arity.h
bc-emit.c
bc-emit.c~
bc-emit.h

8

O.html

Edit Search Project Preferences Help View

```
L PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

"GENERATOR" CONTENT="SGML-Tools 1.0.9">
 IPCHAINS-HOWTO</TITLE>
"HOWTO-1.html" REL=next>

</HEAD>
<BODY>
<A HREF="HOWTO-1.html">Next</A>
Previous
Contents
<HR>
<H1>Linux IPCHAINS-HOWTO</H1>

<H2>Paul Russell, <CODE>ipchains@rustcorp.com</CODE></H2>v1.0.7, Fri Mar 12 13:46:20 CST
1999
<P><HR>
<EM>This document aims to describe how to obtain, install and configure
the enhanced IP firewalling chains software for Linux, and
some ideas on how you might use them.</EM>
<HR>
<P>
```

Line 7 /usr/doc/packages/ipchains/ipchains-HOWTOs-1.0.7/HOWTO.html

Fig. 5: l'about di CodeCrusader.
Fig. 6: un programma C editato con CodeCrusader.
Fig. 7: CodeCrusade gestisce interamente un progetto, automatizzando la creazione di makefile e header.
Fig. 8: CodeCrusader Gestisce anche l'Html.

programmazione avanzata

NETWORK PROGRAMMING CON LINUX

FELICE MAINOLFI
f.mainolfi@oltrelinux.com

25 anni, laureando in Ingegneria Elettronica, si occupa di sistemi distribuiti e sicurezza in ambiente unix.

Il successo di Linux e' dovuto in buona parte alla sua flessibilita' nei confronti della rete. I socket sono uno dei mezzi con cui è possibile interfacciarsi con essa. In questo articolo forniamo una introduzione all'uso dei socket nei nostri applicativi scritti in linguaggio C.

PREREQUISITI:
Conoscenza del Linguaggio C.

Il successo di Linux come sistema operativo finalmente in grado di competerecon Windows è dovuto, in massima parte, alla **flessibilità** con cui si interfaccia con la **rete**. Ogni informazione che circola su internet è suddivisa in una serie di pacchetti organizzati secondo opportuni protocolli. L'accoppiata **TCP/IP** è la più usata per le normali comunicazioni client-server.

La gestione di questi pacchetti è affidata direttamente al **kernel**. L'utente che intenda programmare per la rete potrà servirsi dell' "interfaccia" offerta dai **socket**. In questo articolo vedremo cosa sono e come utilizzarli per costruire in C un client e un server basati sul TCP/IP.

Il modello Client-Server

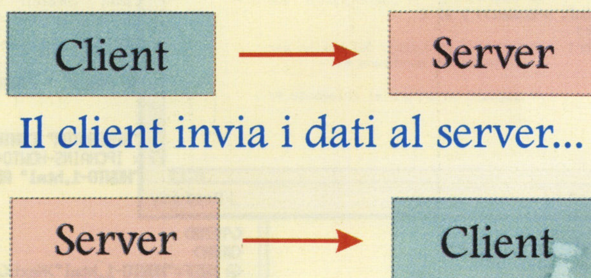
Alla base del funzionamento delle comunicazioni in rete è il modello **client-server**. Facciamo riferimento alla figura 1: il client fornisce dati al server, il server esegue una certa elaborazione e ne invia i risultati al client; sia il client sia il server sono processi che possono risiedere sulla stessa macchina oppure su macchine diverse ma collegate in rete. Questo è il caso, ad esempio, del nostro browser web (client) che invia richieste ad un server http (server).

Dal punto di vista dei socket i compiti che un client deve eseguire sono i seguenti

1) creare un socket
2) attribuire al socket i parametri del server (indirizzo, porta ...)
3) connettersi al server
4) inviare e ricevere dati attraverso il socket
5) chiudere la connessione

Per quanto rigurda il server esso deve essere in grado di gestire anche più **connessioni contemporaneamente**. La via più naturale (*ma non la più "economica" in termini di occupazione di risorse*) è quella di creare tanti processi **figli** del server che soddisfano le richieste dei client collegati in quel momento. Schematicamente un server deve:

1) creare un socket da mettere "in ascol-





```
Color xterm
LFFoundry:~# gcc -o fingerclient fingerclient.c
LFFoundry:~# fingerclient localhost root

Welcome to Linux version 2.2.8 at LFFoundry.GreenNet.com !

  9:55am  up 1:14,  5 users,  load average: 0.09, 0.04, 0.01

Login: root                             Name:
Directory: /root                        Shell: /bin/bash
On since Mon Sep 27 08:41 (CEST) on tty1    1 minute 34 seconds idle
On since Mon Sep 27 08:42 (CEST) on tty2    6 minutes 56 seconds idle
On since Mon Sep 27 08:54 (CEST) on tty3    18 minutes idle
On since Mon Sep 27 09:54 (CEST) on tty4    45 seconds idle
On since Mon Sep 27 09:54 (CEST) on ttyp0 from :0.0
No mail.
No Plan.

LFFoundry:~# 
```

Figura 2: output di fingerclient.c

to"
2) assegnare ad esso una porta
3) definire il numero massimo di connessioni contemporanee
4) porsi in attesa delle connessioni
5) associare un socket ad ogni client che "chiama"
6) creare un figlio per gestire le richieste del client
7) tornare al punto 4

conclusa la comunicazione, al figlio non resta che chiudere il socket associato al client e terminare.

Creare un socket

Un socket, dal punto di vista del linguaggio C, è un descrittore di file. Così come ad un file viene assegnato mediante la funzione fopen() un numero intero che lo identifica (il descrittore, appunto), così un socket è identificato da un numero intero. In luogo di fopen(), per creare un socket si usa la funzione:

```
int socket (int af, int type,
                    int protocol);
```

il risultato è un numero intero che **identifica** il socket creato. Se qualcosa è andato storto ritorna **-1**. L'argomento af specifica il formato degli indirizzi, AF_INET è il valore relativo agli indirizzi internet. Il parametro type definisce il tipo di socket; può assumere sostanzialmete 3 valori **SOCK_STREAM**, per le comunicazioni via TCP, **SOCK_DGRAM** per quelle UDP, **SOCK_RAW** per le situazioni nelle quali è il programma stesso a costruire i pacchetti. Infine protocol identifica il protocollo da usare: se è 0 allora viene scelto quello più adatto in relazione al tipo di socket.
In definitiva per creare un socket per comunicazioni via TCP basta scrivere qualcosa del genere:

```
int s;
...
s = socket (AF_INET,
                    SOCK_STREAM, 0);
```

Il Byte Order

Prima di passare alle funzioni che gestiscono la connessione al server è necessario capire cosa si intende per "**byte order**". I valori numerici su un computer basato su processore Intel sono organizzati in byte ordinati dal **meno** al più significativo (bigendian byte order). L'instradamento dei pacchetti su Internet, invece, richiede che i valori numerici rappresentativi dei parametri di connessione (indirizzo, porta) siano ordinati dal byte **più** significativo a quello meno (network byte order). Processori come quelli della famiglia Motorola 68000 adottano una convenzione uguale a quella di internet.
Per evitare imprevisti nei passaggi da una architettura all'altra (il byte order dell' host può coincidere o meno con il network byte order) vengono messe a disposizione **quattro** funzioni:

```
long ntohl  (long netlong)
long htonl  (long hostlong)
short ntohs  (short netshort)
short htons  (short hostshort)
```

il loro significato è chiaro: htonl, per esempio, converte l'intero lungo hostlong (che è nel formato adottato dalla vostra macchina) in un intero lungo in formato "network".
Normalmente l' indirizzo IP di un host è organizzato su 4 byte ed è memorizzato in un long, mentre il numero di porta è su 2 byte ed uno short è sufficiente a memorizzarlo.

Strutture per l'indirizzamento

L'indirizzo e la porta del server al quale collegarci vengono posti in una apposita struttura:

```
struct sockaddr {
  short sa_family;
  char sa_data[14];
};
```

che, mediante casting, viene particolarizzata per gli indirizzi internet:

```
struct sockaddr_in {
  short sin_family;
  short sin_port;
  struct in_addr sin_addr;
};
```

nella quale la struttura in_addr è semplicemente:

```
struct in_addr {
  long s_addr;
};
```

Nel caso specifico sin_family sarà sempre pari a AF_INET. Un discorso a parte meritano sin_port e sin_addr.s_addr che contengono porta e indirizzo numerico del server al quale connettersi: importante è l'uso delle funzioni per il byte order, come illustrato nel paragrafo precedente.

Interrogare il DNS

Come avete potuto osservare è di fondamentale importanza conosce l'ip numerico di un host quando si va a "riempire" la struttura sockaddr_in. Il più delle volte è noto esclusivamente l'identificativo **alfanumerico**: come è possibile passare dall'uno all'altro? La funzione che fa al caso nostro è:

```
struct hostent *gethostbyname
                    (char *name);
```



Programmazione

che fornisce in **uscita** una struttura hostent contenente, tra le altre cose, anche l'indirizzo numerico del server contenuto in name. La struttura hostent è dichiarata come:

```
struct  hostent {
   char      *h_name;
   char      **h_aliases;
   int       h_addrtype;
   int       h_length;
   char      **h_addr_list;
#define h_addr  h_addr_list[0]
};
```

in particolare ci occorreranno i campi h_lenght e h_addr che contengono rispettivamente la lunghezza e l'indirizzo dei byte contenenti l'indirizzo numerico cercato. Ad esempio, supponiamo di aver bisogno di riempire una struttura sockaddr_in per collegarci all' host tizio.pincopallino.it; il codice che scriveremo sarà il seguente:

```
struct hostent *hp;
struct sockaddr_in address;
...
hp = gethostbyname
         ("tizio.pincopallino.it");
memset(&address,0,sizeof(address));
memcpy((char *)&address.sin_addr,
           hp->h_addr,hp->h_length);
address.sin_family = AF_INET;
address.sin_port = htons(porta);
```

ottenendo la struttura address pronta per essere usata. Un'altra situazione che si verifica spesso è quella in cui a partire da una struttura sockaddr_in, contenente un indirizzo numerico, si vuole risalire a quello alfanumerico; la funzione che useremo sarà:

```
struct hostent *gethostbyaddr(
         char *addr, int len, type);
```

dove addr è una struttura sockaddr_in, len la sua lunghezza e type, come al solito, è AF_INET.

Connettere un socket

Finalmente abbiamo tutto l'occorrente per iniziare la connessione tramite il socket creato con **socket**(). La comunicazione col server viene instaurata tramite la funzione **connect**() :

```
int connect (int s, struct
      sockaddr *name, int namelen);
```

s è il socket da connettere, name una struttura sockaddr_in la lunghezza in byte della quale è contenuta in namelen. Da notare che name è una struttura di tipo sockaddr, cioè una generalizzazione della sockaddr_in usata in ambito internet. Se la connessione non è andata a buon fine otterremo -1 come valore di ritorno.

Inviare dati

Terminate queste fasi preliminari, è possibile iniziare a inviare e ricevere dati tramite il socket. Esistono diverse funzioni preposte allo scopo. Dato che un socket può essere considerato un vero e proprio identificatore di file, si possono usare le funzioni read() e write() per leggere o scrivere:

```
int read(int socket,
         char *buffer, int numbyte);
int write(int socket,
         char *buffer, int numbyte);
```

nelle quali buffer è un puntatore al buffer dei dati e numbyte la sua lunghezza in byte. In uscita avremo il numero di byte letti o scritti o un intero minore di 0 in caso di errore.
Esiste una differenza sostanziale fra l' I/O su socket e su file: nel primo caso non sempre vengono letti o scritti lo stesso numero di caratteri richiesti; così, ad esempio, è necessario usare un loop fino a quando il numero di caratteri desiderato è stato inviato o ricevuto.

```
Color xterm
LFFoundry:~# gcc -o utmpview utmpview.c
LFFoundry:~# utmpview &
[1] 331
LFFoundry:~# telnet localhost 20000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to RemoteUTMP service  Version 0.1.1
--------------------------------------------------
                Mon Sep 27 08:41:16 1999
runlevel        ~    Mon Sep 27 08:41:16 1999
                Mon Sep 27 08:41:26 1999
root     tty1   Mon Sep 27 08:41:33 1999
root     tty2   Mon Sep 27 08:42:59 1999
root     tty3   Mon Sep 27 08:54:04 1999
root     tty4   Mon Sep 27 09:54:11 1999
LOGIN    tty5   Mon Sep 27 08:41:26 1999
LOGIN    tty6   Mon Sep 27 08:41:26 1999
root     ttyp0  Mon Sep 27 11:30:36 1999
Connection closed by foreign host.
LFFoundry:~# █
```

Figura 3: output di utmpview.c

Riquadro 1

Il servizio FINGER

Il servizio finger è uno dei più tradizionali e semplici da usare su un sistema unix. Collegandoci ad esso possiamo ricevere informazioni sugli utenti in quel momento collegati oppure su un user specifico. È anche uno dei servizi che vengono disabilitati per questioni di sicurezza. Se sulla vostra macchina non è attivo basta decommentare la corrispondente
linea in /etc/inetd.conf e costringere inetd al refresh con

```
killall -HUP inetd
```

Potete controllare l'attivazione del servizio col comando

```
netstat -a | grep finger
```

Per capire cosa inviare per ottenere un responso utilizziamo telnet:

```
telnet localhost 79
```

dopo la connessione premiamo invio e si ottiene la lista degli utenti collegati (nel nostro caso, le varie console virtuali e eventuali collegamenti esterni). Ricollegandoci se prima di premere invio inseriamo l'username di un utente, ad esempio root, otterremo maggiori info sull'utente root. Il programma fingerclient.c ottiene il suo output proprio "emulando" una sessione telnet siffatta.



Chiudere un socket

La **chiusura** di un socket è classicamente affidata alla funzione **close**(): un socket viene chiuso in maniera analoga a un file. Tuttavia la funzione

```
int shutdown(int s, int how);
```

consente di chiudere un socket in maniera più specifica: **shutdown**() accetta, oltre al socket s, un ulteriore parametro **how** che specifica in che modo chiuderlo; quest'ultimo parametro può assumere i seguenti valori

- **0** - il socket continua a ricevere ma è chiuso per le operazioni di scrittura;
- **1** - il socket può continuare a trasmettere ma è chiuso in ricezione;
- **2** - il socket viene chiuso completamente

Come al solito il valore in uscita è negativo se si è verificato un errore. Da notare che è comunque necessario ricorrere alla close(), in quanto shutdown() non libera le risorse associate al socket.

Un client minimalista

Passiamo alla pratica. Nel listato 1 è presentato un **client** per **finger** (per i dettagli sul servizio finger vedi riquadro 1). Come potete osservare, sebbene il programma sia molto breve sono presenti tutti i concetti presentati. L'unica "stranezza" è che **non** vengono controllati i byte effettivamente mandati o ricevuti da write() e da read(). Questo a rigore non sarebbe corretto e una soluzione più elegante prevede l'utilizzo di stream associati al socket; tuttavia, dato che i byte in uscita sono pochi e che in ingresso si provvede a una lettura "ciclica", difficilmente vi troverete di fronte a una situazione in cui il meccanismo cada in errore.
Una annotazione importante riguarda i file .h utilizzati:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
```

sebbene ce ne siano molti altri attinenti all'argomento socket questi sono quelli d' uso in una comunicazione basata su socket di tipo SOCK_STREAM. La compilazione e l'uso del client non è difficile e potete vedere un esempio di output in figura 2.

Il Binding

Continuiamo esamindando quelli che sono i concetti necessari a scrivere un server. Alla fase di creazione del socket segue il "**binding**". In pratica al socket viene associata la **porta** sulla quale il server deve mettersi in **ascolto**.
Anche in questo caso è necessario riempire una struttura sockaddr_in prima di passare al binding vero e proprio; tipicamente si procede in questo modo

```
struct sockaddr_in *addr;
...
addr = (struct sockaddr_in *)
  malloc(sizeof(struct sockaddr_in));
addr->sin_family = AF_INET;
addr->sin_port = htons(porta);
addr->sin_addr.s_addr =
                    htonl(INADDR_ANY);
```

a differenza del client stavolta non è necessario specificare un indirizzo numerico (o meglio, va usato quello "speciale" INADDR_ANY). Ciò è naturale in quanto al server è sufficiente conoscere solo su quale porta ricevere le connessioni.
L'associazione tra la porta e il socket viene eseguita con la funzione:

```
int bind(int sockfd, struct
    sockaddr *my_addr, int addrlen);
```

come al solito sockfd è il socket, my_addr è una struttura sockaddr_in inizializzata come appena visto e addrlen è la sua lunghezza.
Un ulteriore passo da compiere è definire il **numero massimo** di **connessioni** contemporanee accettabili. Ciò consente al kernel di dimensionare in maniera corretta la **coda interna** associata al socket. **Listen**() esegue proprio questo compito

```
int listen(int s, int backlog);
```

dove backlog è il numero massimo di connessioni.

Accettare connessioni

...è il compito fondamentale del server. Consideriamo la funzione

```
int accept(int s, struct sockaddr
              *addr, int *addrlen);
```

in essa s è il socket bindato in precedenza mentre addr e addr sono una struttura sockaddr_in e la sua lunghezza. Il valore in uscita è un ulteriore socket, collegato al client che ha richiesto la connessione. Nel nostro caso accept() è una funzione bloccante, nel senso che non restituisce il controllo al programma chiamante fino a che non arriva una richiesta di connessione; una volta ricevuta accept() fornisce un socket collegato al client, e sul quale si possono ricevere ed inviare i dati. La struttura addr contiene i parametri del client come, ad esempio, il suo indirizzo numerico.

Riquadro 2

Il miniserver utmpview

Il file utmp contiene una serie di informazioni sugli utenti in quel momento collegati al sistema. Da questo file attingono programmi come who e finger.
Nel nostro esempio leggiamo i record in esso contenuti mediante l'apposita struttura utmp dichiarata nel file utmp.h. I dati "salienti" di questa vengono poi inviati tramite il socket associato al client nella funzione clientdialog(). Una buona idea, dopo la compilazione, è avviare il server in background con

```
utmpview &
```

Quale client usiamo per collegarci? Dato che il server invia una serie di stringhe di caratteri può bastare telnet.
La porta che abbiamo scelto per il server è la 20000 di conseguenza il suo output può essere esaminato con il comando

```
telnet localhost 20000
```

dove supponiamo che stiate provando il server in locale (altrimenti al posto di localhost dovreste mettere l'indirizzo dell'host dove risiede il server). Un esempio di quello che potete ottenere è riportato in Figura 3.

Un esempio di server

Nel listato 2 è riportato un piccolo server che si limita a fornire ai client un dump del file /var/run/utmp (vedi riquadro 2). Dopo la creazione e il binding del socket associato al server parte quello che è il ciclo principale del server. Quando la accept() fornisce il socket associato a un client il processo si "**clona**" attraverso la fork(). Osserviamo il seguente stralcio di codice:

```
if((procid = fork()) == -1) {
  perror("fork()");
        exit(104);
}
if((int)procid == 0)
        clientdialog(sclient);
else
        close(sclient);
```

procid è il risultato della fork(): se procid è 0 allora ci troviamo nel processo **figlio** il quale si incarica di mandare i dati al client il socket sclient nella funzione clientdialog; se procid non è 0 allora siamo nel processo **padre**, che può chiudere il socket sclient e continuare ad accettare connessioni.

Notiamo che, a differenza di fingerclient.c, la scrittura avviene in maniera più ortodossa mediante la funzione write_data().

Gestione avanzata degli errori

Le situazioni di errore vengono tipicamente segnalate mediante valori negativi nei risultati delle funzioni che hanno provocato l'errore. Un programma un pò più evoluto può tentare di recuperare gli errori a partire da una identificazione più precisa delle cause che li hanno provocati. È possibile ottenere questa informazione includendo il file **errno.h** e esaminando la variabile globale errno. Ogni funzione può produrre un certo insieme di errori, riportati in fondo alle manpage di ognuna.

Conclusioni

Certamente questa rapida escursione nel mondo dei socket **non** può esaurire l'argomento. Tuttavia ora dovreste essere di scrivere i vostri primi **client e/o server**. Non perdete però di vista un punto fondamentale: se non inserite qualche meccanismo di **esclusione** chiunque può accedere al vostro server che magari è vulnerabile a qualche tipo di attacco. Non fornire al server i permessi di **root**, quando possibile, è già una **prima precauzione**, che comunque non mette al sicuro da intrusioni indesiderate. Programmare con un occhio alla **sicurezza** è fondamentale in questi situazioni, non dimenticatelo.



LISTATO 1 (fingerclient.c)

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
int s;
        unsigned short porta = 79;
        struct sockaddr_in address;
        struct hostent *hp;
        char *datainput;

        if (argc!=3) {
        printf("%s <host> <user>\n", argv[0]);
        exit(1);
        }

        /* Creazione del socket */

        s = socket(PF_INET, SOCK_STREAM, 0);
        if (s == -1) {
        perror("socket()");
                exit(1);
        }

        /* Risoluzione dell'ip numerico del server */

        if ((hp = gethostbyname(argv[1])) == NULL) {
                perror("gethostbyname()");
                exit(1);
        }

        /* Inizializzazione della struttura sockaddr_in */

        memset(&address,0,sizeof(address));
        memcpy((char *)&address.sin_addr,
            hp->h_addr,hp->h_length);
        address.sin_family = AF_INET;
        address.sin_port= htons((u_short)porta);

        /* Connessione */

        if (connect(s, (struct sockaddr *) &address,
            sizeof(address)) == -1) {
        perror("connect()");
                exit(1);
        }

        /* Invio e ricezione dei dati finger */

        write(s, argv[2], strlen(argv[2]));
        write(s, "\n", 1);

        datainput = (char *) calloc (80, sizeof(char));
        while (read(s, datainput, 80) != 0) {
        printf("%s", datainput);
        bzero(datainput, 80);
}

        printf("\n");

        /* Chiusura del socket */

        close(s);
 }
```

LISTATO 2 (utmpview.c)

```
#include <stdio.h>
#include <utmp.h>
#include <fcntl.h>
#include <string.h>
#include <signal.h>
#include <sys/socket.h>
#include <netinet/in.h>

int write_data(int s, char *buf, int n)
{
int bcount; /* contatore dei byte scritti */
int br;     /* byte scritti in un ciclo */

        bcount = 0;
        br = 0;
        while (bcount < n) {
           /* loop finchè il buffer non è vuoto */
      if ((br = write(s, buf, n-bcount)) > 0) {
              bcount += br;
                      buf += br;
                      /* muovi il puntatore del buffer
                         per la prossima scrittura */
              }
        else if (br < 0)
                        /* segnala un errore al chiamante */
      return(-1);
        }
        return(bcount);
}

void clientdialog(int sclient)
{
char msg[120];
struct utmp ut;
int fp;

strcpy(msg,"Welcome to RemoteUTMP service  Version 0.1.1\n");
write_data(sclient, msg, strlen(msg));
strcpy(msg,"-------------------------------------------\n\n");
write_data(sclient, msg, strlen(msg));

fp = open (UTMP_FILE, O_RDONLY);

        while (read(fp, &ut, sizeof(struct utmp)) > 0) {
                strcpy(msg, ut.ut_name);
                /* Prepara messaggio in uscita */
        strcat(msg, "\t");
        strcat(msg, ut.ut_line);
        strcat(msg, "\t");
        strcat(msg, ctime(&(ut.ut_time)));

      if (write_data(sclient, msg, strlen(msg)) == -1) {
      perror("write_data()");
      exit(106);
      }
}

close(sclient);
close(fp);
exit(0);  /* Fine del processo figlio */
}

main(int argc, char **argv)
{
struct utmp ut;
int s, sclient, *clientlen, porta = 20000;
struct sockaddr_in *addr, clientaddr;
pid_t procid;

/* Creazione socket; s è il descrittore
   del socket creato */

if ((s = socket(PF_INET, SOCK_STREAM, 0)) == -1) {
perror ("socket()");
exit(100);
}

/* Definizione delle caratteristiche del socket
   mediante bind() */

addr = (struct sockaddr_in *) malloc(sizeof(struct sockaddr_in));
addr->sin_family = AF_INET;
addr->sin_port = htons(porta);
addr->sin_addr.s_addr = htonl(INADDR_ANY);
if (bind(s, (struct sockaddr *) addr,
  sizeof(struct sockaddr)) == -1) {
perror ("bind()");
exit(101);
}

/* Il massimo numero delle connessioni contemporanee è 3 */

if (listen(s, 3)==-1) {
perror ("listen()");
exit(102);
}

/* Antizombie */

signal(SIGCHLD, SIG_IGN);

/* Ciclo principale del server; qui vengono accettate le
   connessione e creati i processi figlio
   che le gestiscono. */

*clientlen = sizeof(clientaddr);

for(;;) {
if((sclient = accept(s, (struct sockaddr *) &clientaddr,
  clientlen)) == -1) {
    perror("accept()");
    exit(103);
  }

/* Creazione del processo figlio mediante fork() */

  if((procid = fork()) == -1) {
    perror("fork()");
    exit(104);
  }

  if((int)procid == 0)
  clientdialog(sclient);
  else
  close(sclient);
}

}
```



Jilani Khaldi - j.khaldi@oltrelinux.com

EVOLUZIONE DELLA SPECIE

HTML → XML

Java/XML - un binomio per il Web

Quando fu inventato il telefono, i più ottimisti e sostenitori di quella nuova tecnologia non avevano minimamente previsto il suo grande successo; pensavano che la sua massima diffusione non sarebbe andata oltre questa previsione: un telefono a metà piani in ogni condominio, usato solo in casi di emergenza. Non avevano assolutamente previsto il fenomeno dei cellulari! In quei tempi, tale idea era semplicemente inconcepibile. Per decretare il successo di una tecnologia, certamente, non bastano **mode** e **marketing**, ma le sue qualità e capacità di risolvere problemi in modo **adeguato**, **semplice**, **poco costoso** e **duraturo**. Le qualità di Java, soprattutto in termini di portabilità e dinamicità, abbinate a quelle di XML (eXtensible Markup Language), sono già in grado di superare molti limiti imposti dalla "vecchia" tecnologia che domina tuttora lo scambio di informazioni tramite Internet. Basti pensare al CGI lato server e a HTML lato client.
La caratteristica principale di XML è quella di permettere all'utente di definire nuovi **tipi di dati** in modo semplice, rigoroso ed elegante. I dati possono spaziare dai semplici testi, alle note musicali, alle equazioni di matematica, alle formule di chimica, ai disegni vettoriali CAD... Queste qualità lo rendono ideale per le pubblicazioni sul Web e lo candidano a diventare il formato standard per lo scambio dei dati.

> La caratteristica principale di XML è quella di permettere all'utente di definire nuovi tipi di dati in modo semplice, rigoroso ed elegante.

Un parser per XML

In realtà, un documento XML è un semplice **contenitore** di una grande varietà di informazioni, che può viaggiare attraverso una rete, passando da un computer ad un altro, con tutto il suo contenuto. Il parser ha il compito di controllare rigorosamente la sintassi del documento, estrarre e manipolare i dati contenuti in esso.
Il parser svolge un ruolo fondamentale sia nella fase di codifica del documento, sia in quella della manipolazione dei dati. Un file XML, non banale, è generalmente **lungo** e **complesso**. Gli errori possono insinuarsi in qualsiasi parte del documento. A differenza di HTML, XML ha una sintassi molto rigorosa e non tollera nessun genere di errori. Anche se col passare del tempo si renderanno disponibili sul mercato molti strumenti sofisticati e visuali (RAD), che agevolano la scrittura dei documenti XML, per iniziare è comunque preferibile usare un semplice **editor** di testi. In questo modo, la complessità e l'espressività del linguaggio vengono messe in grande evidenza.

Un parser XML potrebbe essere scritto in qualsiasi linguaggio di programmazione, ma **Java** sembra essere lo strumento più adatto per tale scopo, vista la sua integrazione con Internet, e quindi anche con XML. Non è un caso, che oggi, i due parser più usati, quelli della Sun e dell'IBM, sono stati scritti in Java.
Anche se attualmente il parser XML è un programma esterno, la Sun ha in programma di includere nelle prossime versioni del JDK un parser sofisticato come parte delle API standard di Java.
Il parser della Sun risulta comunque leggermente più veloce e con più capacità di gestire file di grandi dimensioni rispetto a quello dell'IBM.
Nel nostro esempio, useremo ProjectX Parser della Sun.

Un esempio pratico

Come riferimento ho scelto NetQuiz (l'applicazione C/S per creare e fare i quiz all'interno di una Intranet scolastica, trattata in questo stesso numero), che servirà anche come confronto tra le due tecnologie.

Figura 1

```
DTD quiz.dtd

<!-- Dichiarazione della
struttura dati -->
<!ELEMENT Quiz (Quesito)*>
<!ELEMENT Quesito (NoQuesito,
Domanda, Risposta1, Risposta2,
Risposta3, RispCorretta, Punti,
Commento)>
<!ELEMENT NoQuesito (#PCDATA)>
<!ELEMENT Domanda (#PCDATA)>
<!ELEMENT Risposta1 (#PCDATA)>
<!ELEMENT Risposta2 (#PCDATA)>
<!ELEMENT Risposta3 (#PCDATA)>
<!ELEMENT RispCorretta (#PCDATA)>
<!ELEMENT Punti (#PCDATA)>
<!ELEMENT Commento (#PCDATA)>
]>
```

Un quiz è composto da un numero di quesiti, ognuno dei quali è composto da una domanda con varie risposte, delle quali una sola è corretta e da un punteggio. Cerchiamo di creare un documento XML che rappresenti un quiz.

Un documento XML è generalmente composto da due file, il primo contiene la struttura dei dati, cioè il **DTD** (Data Type Definition), il secondo tutte le altre informazioni. Quando il primo file è di ridotte dimensioni viene inglobato nel secondo per formare un unico documento. La scelta di dividere il documento in due parti rimane del tutto **personale**, dipende dalla convenienza di mantenerle separate o meno. Nel nostro caso, teniamo le due parti **separate** per una migliore comprensione.

Il nostro DTD è definito in figura 1.

La struttura ad albero del DTD è la seguente:

```
Quiz
 |--- Quesito
      |------- NoQuesito
      |------- Domanda
      |------- Risposta1
      |------- Risposta2
      |------- Risposta3
      |------- RispCorretta
      |------- Punti
      |------- Commento
```

Figura 3

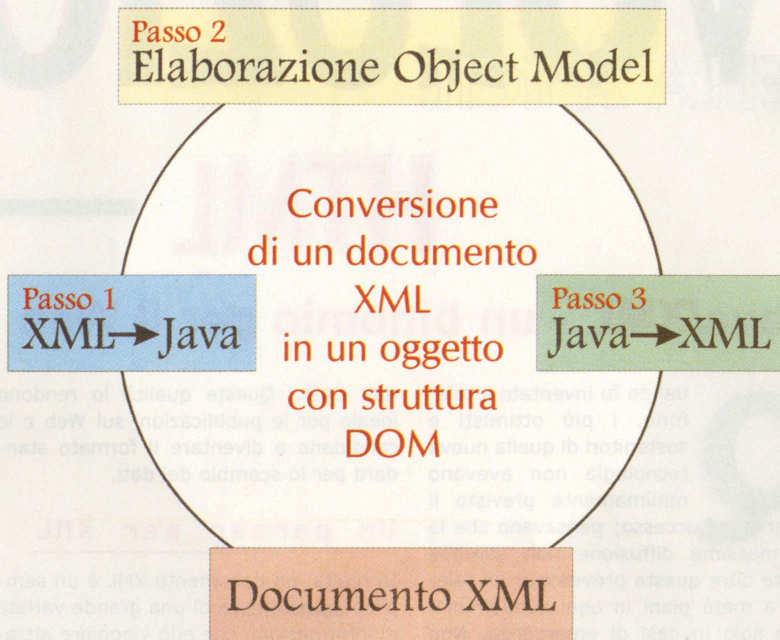


Figura 2

```
netquiz.xml

<?xml version="1.0"?>
<Quiz>
<Quesito>
<NoQuesito> 1 </NoQuesito>
<Domanda> La cupola del duomo di Firenze fu realizzata da:
</Domanda>
<Risposta1> Michelangelo Buonarroti</Risposta1>
<Risposta2> Filippo Brunelleschi </Risposta2>
<Risposta3> Domenico Fontana </Risposta3>
<RispCorretta> 2 </RispCorretta>
<Punti> 1 </Punti>
<Commento> La cupola del Duomo di Firenze fu progettata e
realizzata dal Brunelleschi </Commento>
</Quesito>
<Quesito>
<NoQuesito> 2 </NoQuesito>
<Domanda> Quale imperatore romano ordinò la realizzazione del
Pantheon? </Domanda>
<Risposta1> Augusto</Risposta1>
<Risposta2> Claudio </Risposta2>
<Risposta3> Adriano </Risposta3>
<RispCorretta> 3 </RispCorretta>
<Punti> 1 </Punti>
<Commento> La costruzione del Pantheon fu ordinata da Adriano e
dedicata a tutte le divinità </Commento>
</Quesito>
</Quiz>
```

Il file XML corrispondente lo potete osservare in *figura 2*.

In questo caso, il nostro documento contiene solo due record. La metodologia resta invariata anche se il numero dei record aumenta considerevolmente. La funzione del parser ora, oltre il naturale controllo della sintassi, è quella di creare un oggetto con struttura **DOM** (Document Object Model) a partire dal documento XML,.

DOM è stato definito dal W3C (World Wide Web Consortium) come un insieme di interfacce, indipendenti dalle piattaforme e dai linguaggi, che permette di accedere e modificare i dati e la struttura di un documento XML.

Possiamo riassumere graficamente, ed in seguito in dettaglio, i tre passi necessari per completare l'operazione di **parsing**, **estrazione** e **manipolazione** dei dati da un file XML (*figura 3*).

Si vede chiaramente come l'elemento Quiz viene mappato nella classe Quiz e



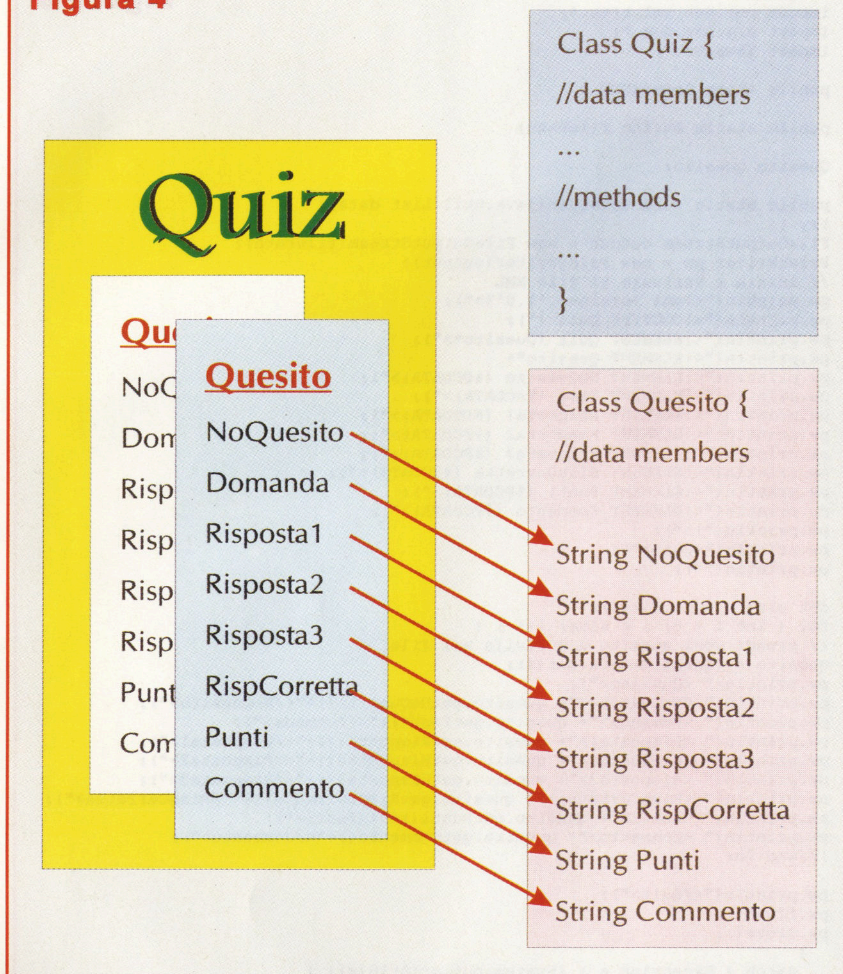


Figura 4

l'elemento Quesito nella classe Quesito. La classe Quiz è semplicemente un contenitore degli oggetti Quesito (figura 4). Per creare un documento DOM a partire dal nostro file XML ci serviamo del seguente codice:

```
try {
 String xmlquiz="quiz.xml";

//  file xml da aprire

 InputStream is= new
 FileInputStream (xmquiz);

// converte il file in uno stream
di input

 Document doc= XmlDocumentBuilder.
 createXmlDocument(is);

// crea il documento

 }
 catch (Exception e)

// segnala gli errori

}
```

Per estrarre i dati del primo del documento XML, ci serviamo di questi metodi della classe XmlUtils:

```
Document doc;
int primoRecord = 1;
public static final String
 ROOT_ELEMENT_TAG = "Quesito";

int numeroQuesiti =
 XmlUtils.getSize
 (doc, ROOT_ELEMENT_TAG);

Element Quesito =
  XmlUtils.getElement
  (doc, ROOT_ELEMENT_TAG ,
  primoRecord);
String NoQuesito =
  XmlUtils.getValue(Quesito,
  "NoQuesito");
String Domanda =
  XmlUtils.getValue(Domanda,
  "Domanda");
String Risposta1 =
  XmlUtils.getValue(Quesito,
  "Risposta1");
String Risposta2 =
  XmlUtils.getValue(Quesito,
  "Risposta2");
```

```
public class Quesito{
   protected String NoQuesito, Domanda,
Risposta1, Rispost2, Risposta3, RispoCorretta,
Punti, Commento;
   public Quesito(){}// costruttore di default
   public  void setNoQuesito(String s) {
       NoQuesito = s;
   }
   public  void getNoQuesito {
       return NoQuesito;
   }

   public void setDomanda(String s){
       Domanda = s;
   }
  public void getDomanda{
       return Domanda;
   }

   public void setRisposta1(String s){
       Risposta1 = s;
   }
   public void getRisposta1{
       return Risposta1;
   }

   public void setRisposta2(String s){
       Risposta2 = s;
   }
   public void getRisposta2{
       return Risposta2;
   }
   public void setRisposta3(String s){
       Risposta3 = s;
   }
   public void getRisposta3{
       return Risposta3;
   }
   public void setRispCorretta(String s){
       RispCorretta = s;
   }
   public void getRispCorretta{
       return RispCorretta;
   }
   public void setPunti(String s){
       Punti = s;
   }
   public void getPunti{
       return Punti;
   }
   public void setCommento(String s){
       Commento = s;
   }
   public void getCommento{
       return Commento;
   }

 }// fine della class Quesito
```

```
String Risposta3 =
  XmlUtils.getValue(Quesito,
  "Risposta3");
String RispCorretta =
  XmlUtils.getValue(Quesito,
  "RispCorretta");
String Punti =
  XmlUtils.getValue
  (Quesito, "Punti");
String Commento =
  XmlUtils.getValue(Quesito,
  "Commento");
```

Passo 2: Conversione di una struttura DOM in un oggetto Java

Ora, per poter manipolare questi dati, creiamo un classe Java col nome "Quesito" che abbia la stessa struttura dati di quello contenuto nel documento XML (figura 5).
In seguito, i dati all'interno del quesito vengono modificati col seguente frammento di codice:

```
{
Quesito quesito = new Quesito();
quesito.setNoQuesito(NoQuesito);
quesito.setTitolo(Titolo);
quesito.setRisposta1(Risposta1);
quesito.setRisposta2(Risposta2);
quesito.setRisposta3(Risposta3);
quesito.setRispCorretta(RispCorretta
);
quesito.setPunti(Punti);
quesito.setCommento(Commento);
}
```

A questo punto, abbiamo trasformato il nostro quesito da documento XML ad un oggetto Java, al quale possiamo applicare qualsiasi genere di elaborazione.

Passo 3: Conversione di un oggetto Java in un oggetto XML

Per estrarre i dati dal nostro oggetto Quesito, basta richiamare i suoi metodi get:

```
Quesito quesito = new Quesito();
```

Figura 6

```
import com.sun.xml.tree.*;
import org.w3c.dom.*;
import java.io.*;

public class GeneraXML {

public static String filePath;

Quesito quesito;

public static void saveQuiz(java.util.List data){
try {
FileOutputStream output = new FileOutputStream(filePath);
PrintWriter pw = new PrintWriter(output);
// inizia a scrivere il file XML
pw.println("<?xml version= '1.0'?>");
pw.println("<!DOCTYPE Quiz [");
pw.println("<!ELEMENT Quiz (Quesito*>");
pw.println("<!ELEMENT Quesito"+
pw.println("<!ELEMENT NoQuesito (#PCDATA)>");
pw.println("<!ELEMENT Test (#PCDATA)>");
pw.println("<!ELEMENT Risposta1 (#PCDATA)>");
pw.println("<!ELEMENT Risposta2 (#PCDATA)>");
pw.println("<!ELEMENT Risposta3 (#PCDATA)>");
pw.println("<!ELEMENT RispCorretta (#PCDATA)>");
pw.println("<!ELEMENT Punti (#PCDATA)>");
pw.println("<!ELEMENT Commento (#PCDATA)>");
pw.println("]>");
pw.println("<Quiz>");
pw.println("");

int size = data.size();
for ( int i = 0; i < size; i++ ) {
// prendi ogni quesito e scrivilo sul file
quesito = (Quesito)data.get(i);
pw.println(" <Quesito>");
pw.println(" <NoQuesito>"+ quesito.getNoQuesito()+"</NoQuesito>");
pw.println(" <Domanda>"+ quesito.getTest()+"</Domanda>");
pw.println(" <Risposta1>"+ quesito.getRisposta1()+"</Risposta1>");
pw.println(" <Risposta2>"+ quesito.getRisposta2()+"</Risposta2>");
pw.println(" <Risposta3>"+ quesito.getRisposta3()+"</Risposta3>");
pw.println(" <RispCorretta>"+ quesito.getRispCorretta()+"</RispCorretta>");
pw.println(" <Punti>"+ quesito.getPunti()+"</Punti>");
pw.println(" <Commento>"+ quesito.getCommento()+"</Commento>");
}//end for

pw.println("</Quiz>");
pw.flush();
pw.close();
  }
  catch ( Exception e ) {System.out.println(e); }
}//end method

}//end of Genera XML
```

```
String NoQuestio =
  quesito.getNoQuesito();
String Titolo  =
  quesito.setTitolo();
String Risposta1 =
  quesito.setRisposta1();
String Risposta2 =
  quesito.setRisposta2();
String Risposta3 =
  quesito.setRisposta3();
String RispCorretta =
  quesito.getRispCorretta();
String Punti =
  quesito.getPunti();
String Commento =
  quesito.getCommento();
```

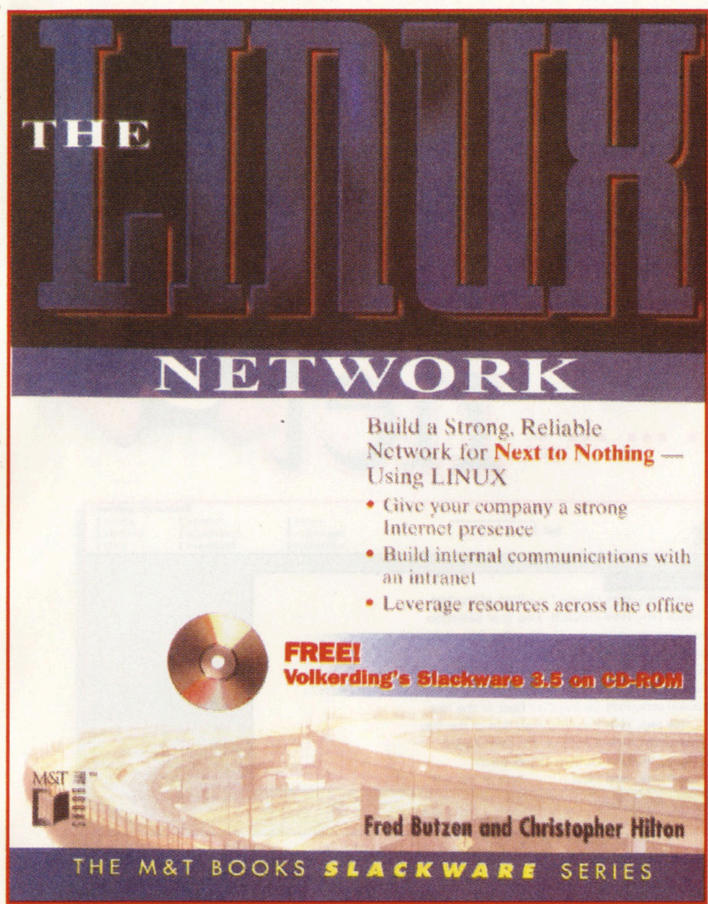
Ecco, in figura 6, un frammento di codice Java per creare il file XML a partire dall'oggetto java.

Come ho accennato nell'introduzione, un documento XML potrebbe contenere una **grande quantità** e **varietà** di dati. Infatti, abbiamo visto come è facile trasformare le informazioni contenute nel documento in oggetti Java e viceversa. A questo punto, lascio immaginare a voi cosa si potrà fare con questa tecnologia in un **futuro non molto lontano**. Molto probabilmente, il binomio **Java/XML** sarà lo strumento **principe** per lo sviluppo e la pubblicazione dei dati del Web che verrà.



Recensione libri





The Linux Network

Casa editrice: M&T BOOKS
Autori: Fred Butzen e Christopher Hilton
Pagine: 526
Prezzo: US $ 39.99
Cd-Rom: Distribuzione Linux Slackware 3.5 (Kernel 2.0.33)
Target: Base - Intermedio.

E' un libro semplice da leggere, ben organizzato e preciso nelle spiegazioni, pratico per implementare tutte le soluzioni in esso contenute, malgrado la complessità e la difficoltà degli argomenti.

A chi si rivolge questo libro? A tutti quelli che vogliono far fare a Linux il suo vero lavoro. Cioè il server di rete TCP. Infatti, questo libro spiega, in dettaglio, come creare una Intranet interamente basata su Linux lato server e con client Windows 95/98/NT a quanti finora non hanno mai trovato né il coraggio di farlo e neppure una guida adatta, come questo libro, per iniziare la grande impresa.

La prima parte, di 68 pagine, è un'introduzione al networking, tratta un po' di tutto: dai protocolli di comunicazione, maggiormente centrato sul TCP/IP, ai vai tipi di reti, di schede, SLIP, PPP, indirizzi, domini... la spiegazione è molto chiara e scorrevole.

La seconda parte, di ben 200 pagine, intitolata "Adding Networking to Your Linux System Hardware", come indica il titolo, riguarda la preparazione dell'hardware per l'allestimento di una rete: installazione della scheda, la scelta delle opzioni del kernel per una compilazione adatta al networking, configurazione del modem e modifica di tutti i file per rendere il nostro Linux un vero server UNIX.
Un ampio spazio è dedicato a chi vuole collegarsi ad Internet con Linux spiegando il tutto in modo molto dettagliato. Segue, poi, l'uso dei programmi classici per il networking sotto UNIX: ftp, telnet, trn, lynx... Una buona parte è dedicata al mailing, e qui non poteva mancare lo standard per tale servizio: sendmail. Infatti, una piccola parte è riservata al suo uso e alla sua configurazione.

La terza, di 112 pagine, intitolata "Creating an Intranet" mantiene veramente la promessa. Inizia spiegando come collegare più macchine tra loro, adottando varie soluzioni, dà altre indicazioni sulla compilazione di un nuovo kernel e su come provare e fare il debug della nuova rete. Prosegue spiegando come preparare i servizi per la nostra rete, con la configurazione dei seguenti server: inetd, Apache, ftpd, telnetd e come condividere le risorse del sistema includendo NFS e Samba.
Una parte è dedicata al masquerading (possibilità di accesso ad Internet da parte di vari client, sfruttando una singola connessione Internet e un solo modem), molto chiara e precisa.

La quarta ed ultima parte, di 104 pagine, spiega in modo esauriente, come collegare un client Windows 95 al nostro server Linux e come condividere tutte le risorse di rete (stampanti, file...).

PRO

Sinceramente, su argomenti ostici, come il networking sotto UNIX, ho trovato pochi libri così chiari, semplici e precisi. Non solo per spiegazione, ma soprattutto per fattibilità. Il tutto concentrato in meno di 500 pagine. Credo che The Linux Networking non debba mancare a chiunque voglia realizzare una Intranet basata su Linux, o anche solo far connettere più PC ad Internet sfruttando una singola connessione. Come nel caso delle scuole, per esempio.
Il libro è dotato di un CD ROM (funzionante!) con la distribuzione Slackware 3.5 con kernel 2.0.33. Ho installato questa distribuzione, su un PC che uso normalmente per le prove, e tutto ha funzionato senza problemi, compresi gli esempi e le indicazioni riportate sul libro.

CONTRO

Se vogliamo cercare il pelo nell'uovo, posso dire che il libro si rivolge a chi ha già una certa familiarità con Linux. Infatti, parte dal presupposto che Linux sia già stato installato e che l'utente abbia le conoscenze basilari per usarlo come worksation. Quindi, con conoscenze sul come modificare i file, crearli, copiarli, attribuire loro i permessi... Altro non ho trovato.





Marco Ivaldi - raptor - m.ivaldi@oltrelinux.com

www.freshmeat.net



Questo mese vi presentiamo la recensione di quello che ormai è diventato uno dei siti "culto" per la comunità Linux. Freshmeat (http://www.freshmeat.net) è un archivio in cui potete trovare ogni giorno le ultime novità software per questo sistema operativo sempre in via di sviluppo, oltre alle ultime notizie che riguardano tutta la comunità che vi ruota attorno.

Freshmeat, prima di diventare un punto di riferimento per tutti gli utenti e sviluppatori di Linux sparsi per il mondo, era semplicemente una spoglia curatore principale Patrick Lenz (meglio conosciuto come scoop), questa veloce e assolutamente priva di grafica pagina HTML era utilizzata esclusivamente ad uso personale per tenere traccia di tutte quelle nuove applicazioni che cominciavano ad essere create per questo sistema operativo ancora relativamente giovane. A mano a mano che l'interesse nella sua piccola pagina cresceva, scoop ha cominciato a pubblicare e a gestire una page HTML statica sul server del suo ISP, aggiornata periodicamente a mano.

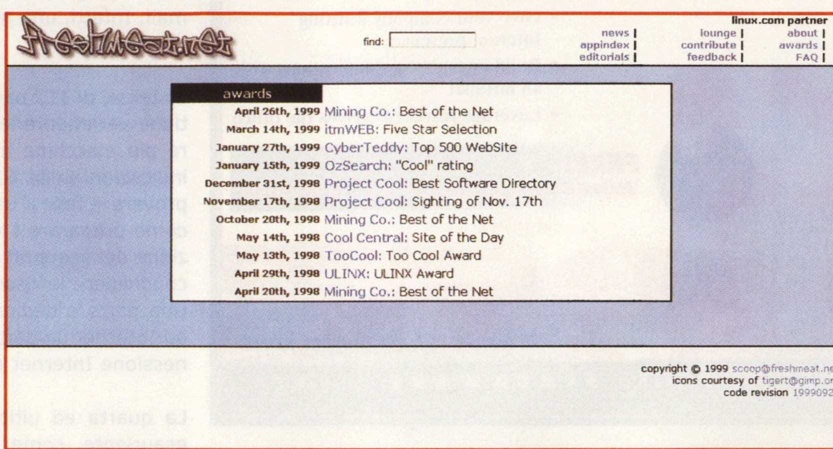


Figura 2: i premi vinti dal sito di freshmeat!

Il PHP: la rivoluzione di freshmeat!

Ciò che cambiò drasticamente l'aspetto e l'importanza di freshmeat fu l'introduzione del PHP (http://www.php.net) e dei primi script per una gestione e un aggiornamento più efficienti. Poco tempo dopo, il sito divenne ufficialmente freshmeat.net e assieme al nome definitivo cominciarono ad arrivare i primi contributi della gente che in tutto il mondo annunciava applicazioni nuove o aggiornate. Vari cambi di aspetto si sono susseguiti nei mesi

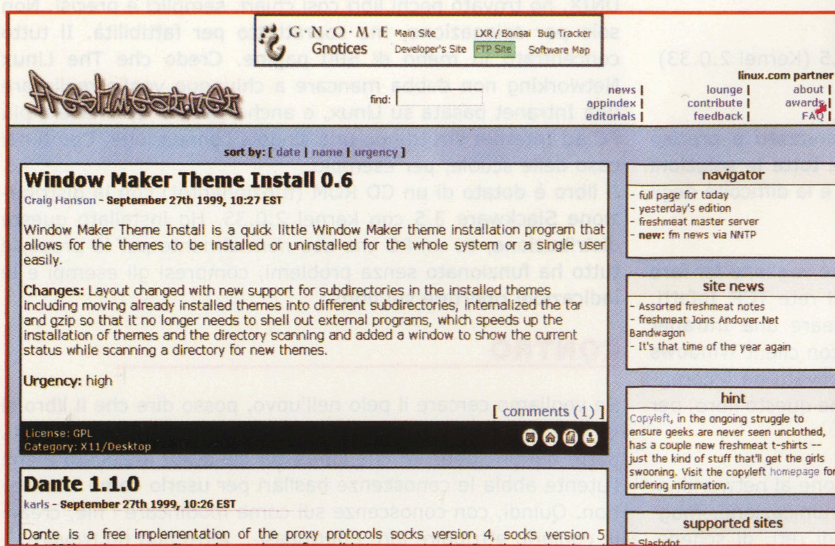


Figura 1: la homepage del sito in tutto il suo splendore...

Il sito WEB del mese



seguenti, accompagnati dall'introduzione di nuove features e tool per la consultazione del sito, come la potente search engine php3 che ora potete utilizzare per trovare velocemente e comodamente qualunque cosa di cui abbiate bisogno. Cambiamenti che hanno portato Freshmeat a ciò che ò ora - un sito decisamente molto visitato con un enorme database di software sempre aggiornato, in cui è possibile trovare ogni tipo di applicazione Linux, per quasi ogni scopo pensabile.

Cosa si puo' trovare?

Si spazia dai demoni alle applicazioni per X11, dai programmi per console ai client per irc, dai tool di sviluppo alle applet per GNOME. Vi è inoltre una grande sezione dedicata all'auditing delle reti e alla sicurezza. Ma Freshmeat non è solamente un database di apps Linux. Vi è anche una sezione News curata personalmente da scoop (scoop@freshmeat.net), con tutte le novità del settore, gli accordi commerciali, le nuove mailing lists e progetti che nascono ogni giorno in questa dinamica comunità. Oltre alla sezione news, possiamo trovare i mirror di due delle maggiori distribuzioni Linux del pianeta: Debian GNU/Linux e Red Hat Linux. La parte Debian è mantenuta da stu (Robert Edmonds), mentre Obituary (Christoph Thompson) e fryguy (Ryan Weaver) si occupano della sezione dedicata a Red Hat. Non va inoltre dimenticata la sezione editoriali, che purtroppo in questo momento non viene più aggiornata, ma i vecchi articoli sono comunque molto interessanti. La forza di Freshmeat è che tutti possono crearsi un account gratuito e segnalare il proprio software e il database sarà aggiornato generalmente in giornata. La grafica è piuttosto essenziale e in generale rende il sito apprezzabile e di facile consultazione per tutti. Sono inoltre disponibili numerosi tool che permettono di visualizzare sul proprio desktop le ultime novità prese direttamente da Freshmeat, o di aggiornare periodicamente la propria home page con le ultime novità dal sito (una di queste è ad esempio il DailyUpdate) Inoltre molti altri siti fulcro di Internet (tra cui my.netscape.com e my.justlinux.com) offrono la possibilità di vedere segnalate le ultime applicazioni e notizie di Freshmeat nelle proprie pagine personalizzate. Insomma, un sito che tutti gli amanti di Linux conoscono e consultano giornalmente per mantenersi sempre informati, vincitore tra l'altro di molti premi, come potete vedere nella pagina degli awards: http://www.freshmeat.net/awards.php3. Al momento della scrittura di questo articolo, tanto per rendere l'idea, il database risulta così composto:

news items: **15473**
appindex records: **4849**
lounge members: **37242**
comments: **3831**
editorials: **31**

Mentre vi godete gli screenshots, vi lascio con un consiglio: visitatelo personalmente se ancora non lo conoscete e mantenetevi sempre aggiornati sul mondo Linux.



YOUR RESOURCE FOR OPEN SOURCE.
find:
news | appindex | editorials | lounge | contribute | feedback | linux.com partner about | awards | FAQ |

sections
[042] Console
[021] Daemons
[018] Development
[010] GNOME
[012] KDE
[007] Web
[047] X11

recent items
- mbi
- secret-share
- radiusd-sql
- whats
- Black Cat Linux
- HDBENCH clone
- TkBible
- InterfaceLED
- HTML::EP::Explorer
- HTML::EP::Glimpse

Figura 3: l'indice della sezione apps (applications)

JUST linux
The complete Linux Guide
Is it harder to remember your homepage URL than your PGP key?
| HOME | My Control Panel | Contact Us |
HELP
If you are not a member of justlinux.com CLICK HERE to register.
If you have forgotten your password CLICK HERE or send an email to help@justlinux.com
LOGIN TO justlinux.com
Username:
Password:
Login Reset
Copyright © 1999 UserChoice.com. All rights reserved. Legal Notice - Privacy Policy
Comments owned by posters. Linux ® is a trademark of Linus Torvalds. Contact us at: support@justlinux.com

find:
news | appindex | editorials | lounge | contribute | feedback | linux.com partner about | awards | FAQ |

step 1/3
The completion of this form submits (after a preview) an article to the freshmeat news database. The **details** input should be used for text explaining the article more detailed. It will only show up on the detailed article pages, never on the frontpage. Hit the **Go** button when you're done.

Please note: Do **not** use this submission method to announce releases of new software. Follow the instructions at the contribution homepage. Thank you.

topic: Example: PHP [help]
body:
details:

Figura 4: la pagina per contribuire all'aggiornamento del sito.

to be or not to be

Ivan Trevisan - magilla - i.trevisan@oltrelinux.com

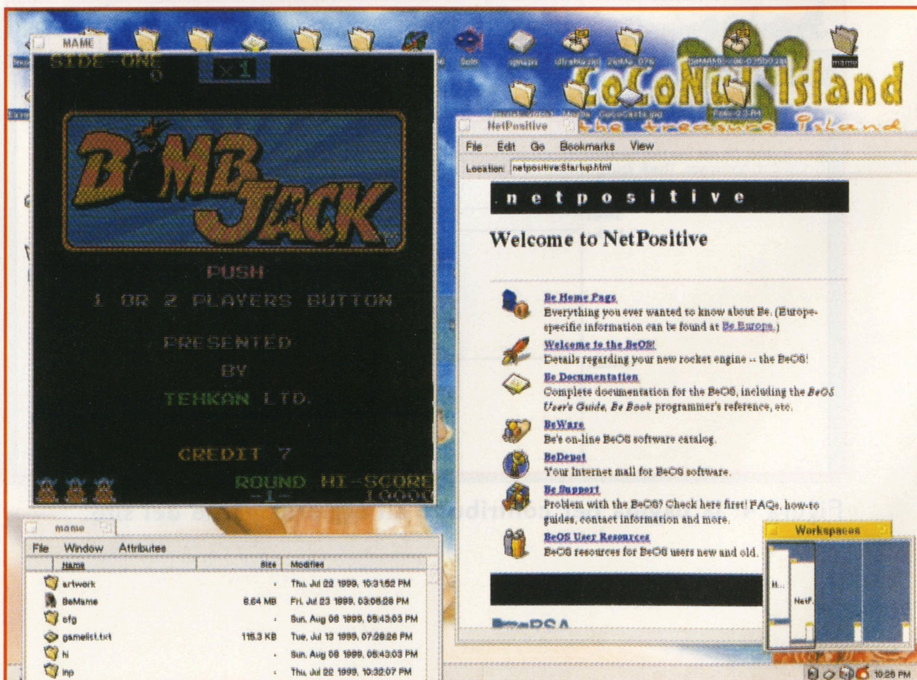
Al giorno d'oggi, il PC risulta essere la macchina più versatile che si possa acquistare, e non solo in virtù dello smisurato parco software, ma soprattutto grazie alla sua capacità di utilizzare i più disparati **sistemi operativi**. Non è una novità dire che il monopolio di mercato è saldamente nelle mani della Microsoft di Bill Gates. Qualsiasi PC prodotto oggi, ha preinstallato Windows98 nelle macchine di fascia bassa e Windows NT Workstation in quelle d'uso professionale. Completa la linea WindowsNT Server, oramai diffusa alternativa ai vari Unix e Netware, nel campo della macchine destinate ad essere utilizzate come server di rete.
Gli utenti più smaliziati, hanno da tempo abbracciato anche la via di Linux, una delle varianti di Unix, creatura sorta dal progetto di un solo uomo, Linus Torvald, che sta salendo al ruolo di "mito" nella comunità di Internet. Linux deve la sua fama grazie all'incredibile supporto e sviluppo che ha ricevuto dalla comunità di entusiasti sviluppatori in Internet. Linux è un software free, pur rientrando nei limiti imposti dalla licenza GNU.
In questo panorama, c'è da chiedersi se ci sia effettivamente **spazio** (e **bisogno**) di un **altro sistema operativo**.
Alla Be Inc. sono certamente convinti che sia questo il caso.

L'era di Windows

Fin dalla sua prima apparizione nel 1995, Windows fece parlare di se per quello che effettivamente fu un balzo in avanti per il mondo dei PC, ma anche un'evidente insieme di soluzioni software per cercare sia di mantenere la **compatibilità** con prodotti software e hardware più vecchi, sia di creare un sistema operativo **moderno** e più **funzionale**. A quattro anni di distanza, Windows non è poi cambiato di molto. Abbiamo ancora il DOS che si nasconde sotto all'interfaccia grafica, la mancanza di un ambiente multiutente, una pessima gestione della memoria, e tanti altri difetti, sotto gli occhi di chiunque abbia lavorato usando pesantemente questo sistema.
Non parliamo poi dei frequenti **blocchi del sistema**; un'applicazione che va in crash, ha **buone** possibilità di riuscire a bloccarvi **l'intera** macchina o perlomeno di forzare il riavvio del sistema.

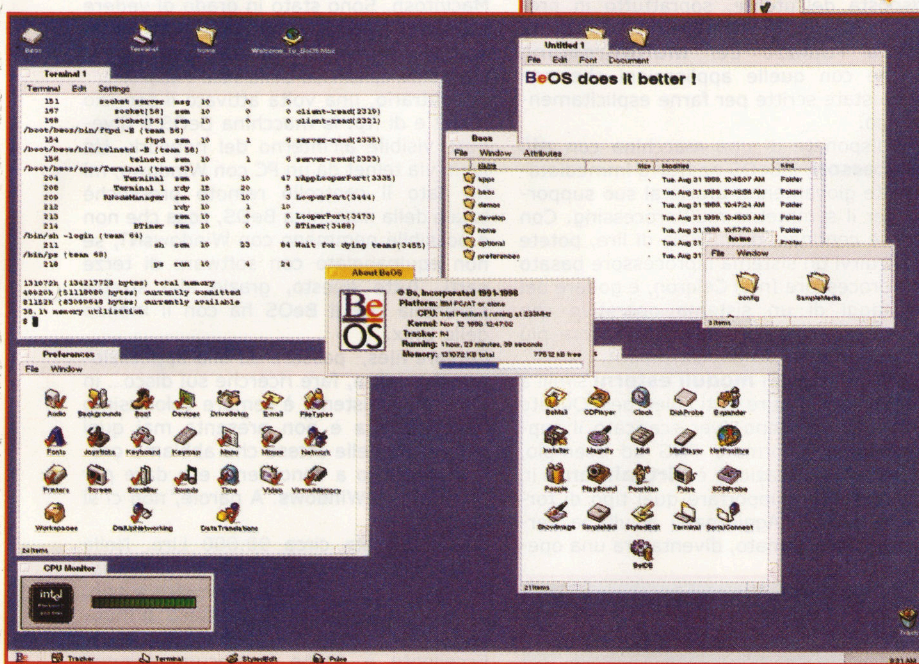
BeOS e la promessa di un MediaOS

Nel 1990 Jean-Louis Gassée lascia il suo posto alla Apple, per fondare una nuova società che ha per scopo lo sviluppo di un nuovo sistema operativo, snello, veloce e in grado di sopperire alle mancanze degli altri OS nel mercato. Nasce così la Be Inc., e qualche tempo dopo, la prima versione del BeOS.
Inizialmente, il BeOS era disponibile solo su piattaforma **Macintosh**. Questo fatto si spiega sapendo che Gassèe ha lavorato per anni alla Apple, e concorda con le voci che volevano il BeOS come successore del MacOS, prima che la Apple decidesse di acquistasse il NextStep, e di modificarlo per farlo diventare il successore del MacOS.
Fu così che a partire dalla release 3 del BeOS, la Be decise di produrre anche una versione Intel compatibile. Molta strada è stata fatta da allora, soprattutto per quanto riguarda il **supporto hardware**, e la possibiltà di **coesistere** con altri sistemi operativi.
Il BeOS infatti, non è mai stato posizionato come un **successore** o un **sostituto** di Windows, ma bensì come un prodotto in grado di affiancarsi e sopperire alle mancanze dell'OS di casa Microsoft.
Il concetto di **MediaOS** deriva dalla filosofia con cui il BeOS è stato concepito, e cioè come prodotto in grado di manipolare grandi quantità di informazioni audio/video, in tempo reale.
Su un Pentium200 MMX, con un programma shareware, sono stato in grado di suonare 3 MP3 contemporaneamente, senza che il sistema diventasse instabile o lento nella risposta, utilizzando una comunissima SoundBlaster16. Il 3D demo poi, convincerà anche il più scettico dei detrattori del BeOS. Si tratta di una applicazione che mappa in tempo reale,





sei filmati AVI (o altro, a scelta) sulle facce del cubo, e ne fa il play mentre voi ruotate il cubo con il mouse.
Il sistema, anche se messo sotto stress,



risponde sempre prontamente, e non ho mai dovuto effettuare un **reboot** forzato anche nei casi più intricati. Fortunatamente poi, la **shell** Unix è sempre in **background**, e un kill dell'applicazione che non risponde risolve praticamente qualsiasi problema. Dalla versione 4.5, il kill dispone di una propria GUI, richiamabile con la pressione dei tasti CTRL-ALT-CANC, facilitando questa operazione a tutti quelli che non hanno confidenza con i comandi della shell.

BeOS nel vostro PC

Se avete intenzione di provare il BeOS, vi consiglio prima di leggere con attenzione la lista dei prodotti compatibili che trovare sul sito della Be (www.be.com). È forse questo infatti, il **problema** che ancora affligge questo nuovo sistema operativo, e cioè la mancanza di un forte **supporto** per **l'immenso** parco hardware disponibile agli utenti PC. C'è da dire che la Be ha impegnato grosse risorse per risolvere questa mancanza, e nel susseguirsi delle versioni (attualmente siamo alla 4.5) la situazione è decisamente migliorata. Si può dire che **quasi sicuramente** il BeOS si installerà su una qualsiasi macchina che abbia hardware non più vecchio di un **anno**, anno e mezzo.
Se alcuni di voi sono preoccupati del fatto di dover installare un altro sistema operativo sul proprio PC, vi assicuro che l'operazione è veramente delle più semplici. Il BeOS viene distribuito con una versione lite di **Partition Magic**. Se avete abbastanza spazio libero, sarà possibile creare una nuova partizione sul vostro disco, utilizzando lo spazio libero rimanente, senza perdere alcun dato. La soluzione migliore rimane sempre un secondo disco, anche di modeste dimensioni (il BeOS non è esoso come Windows in fatto di spazio), su cui installare il nuovo OS.
BeOS 4.5 risiede ora su un **CD bootabile**; questo elimina la necessità di avere un floppy di installazione, come era sempre stato necessario nelle versioni precedenti, e facilita ancor più le operazioni. Nel giro di **dieci minuti**, anche installando il software opzionale, la vostra macchina disporrà di un nuovo OS, finalmente al passo con i tempi !
L'utility di installazione è **semplificata al massimo**, e forse fin troppo per quegli utenti **esperti** che preferiscono avere il controllo totale sulle **opzioni** di **installazione**.
Alla fine dell'installazione, il BeOS vi chiederà se desiderate installare un **boot manager**. Tale operazione si è dimostrata semplice e affidabile, e non ha dato problemi anche in presenza del boot manager di WindowsNT. Maggiore attenzione dovrebbero invece fare quegli utenti che hanno anche Linus installato, non avendo potuto testare se il boot manager del BeOS interferisca in qualche maniera con il LILO di Linux.
In ogni caso, viene data la possibilità di salvare il contenuto del blocco zero sia sul vostro Hard Disk, che su un floppy, per poterlo ripristinare in qualsiasi momento nel caso si presentino dei problemi con il vecchio OS presente nella macchina.
Una volta riavviato il computer, vi si presenterà un menù testuale da cui sarà possibile scegliere quale sistema operativo caricare, ovviamente tra quelli installati e riconosciuti. Dal momento in cui scegliete il BeOS e quello in cui il sistema operativo sarà caricato e pronto ad eseguire i vostri comandi, non passano che **poche manciate di secondi** ! In caso di problemi è sempre possibile premere la barra spaziatrice durante la fase iniziale del boot per entrare in un menù che vi permetterà di scegliere alcune opzioni, tra cui un "safe mode" del tutto simile alla "modalità provvisoria" di Windows.

Prime impressioni

Subito dopo l'avvio, vi verrà presentato il desktop, su cui appare l'icona del disco di boot e del cestino. In alto a destra (ma e' modificabile a piacere), vi e' l'equivalente del menù Start di Windows, contenente tutte le applicazioni installate e le preferenze del sistema. Il look del BeOS è molto **pulito** e decisamente **simpatico**. Non mancherete di sentirvi subito a casa, visto che tutte le operazioni che potete eseguire sono simili a quelle di un qualsiasi ambiente grafico.
Una distinzione importante da fare è che, pur basandosi su un kernel derivante dallo Unix, il BeOS non ha **Xwindow** come interfaccia, ma una vera e propria **interfaccia grafica**, personalizzabile per

alcuni aspetti, ma non modificabile o addirittura sostituibile come è invece possibile fare in Linux.
Sono possibili le classiche viste per icona o per testo, e tutto il sistema è configurabile nella scelta del font e della sua dimensione, degli sfondi del destop, combinazioni di suoni, screensavers, etc. etc.
Già a partire dalla release 4, è stato poi introdotto un metodo di **task switching** del tutto simile a quello utilizzato da Windows. Questo vi permetterà di passare da una applicazione all'altra premendo la classica combinazione ALT-TAB o volendo di passare ad una specifica finestra aperta appartenente all'applicazione in cui vi state spostando.
Merita menzione anche il supporto per le fonts **TrueType**. Questo significa che tutti i caratteri che siete abituati ad utilizzare sotto Windows, funzioneranno felicemente anche con il BeOS.
Da poco, anche il famoso **TuCows** (www.tucows.com) offre una sezione dedicata al BeOS.
In generale, il look'n'feel del BeOS si posiziona a metà via tra Windows e il MacOS, prendendo probabilmente il meglio di entrambi. Sicuramente basteranno pochi minuti per ambientarsi al nuovo sistema operativo.

Pervasive multithreading e symmetric multiprocessing

Queste due sigle che ai più potranno sembrare criptiche e senza significato, sono alla base della fama del BeOS. Con **pervasive multitrheading** si intende la capacità del SO di spezzettare qualsiasi compito in **sottoprocessi**, in modo da mantenere tutto il sistema più veloce alla risposta dell'utente, soprattutto in presenza di grossi carichi di lavoro. Il BeOS **forza** l'utilizzo del **multithreading** anche con quelle applicazioni che non sono state scritte per farne esplicitamente uso.
Se disponete di una macchina con **più processori**, il BeOS ne trarrà immediatamente giovamento, grazie al suo supporto per il symmetric multiprocessing. Con poche centinaia di migliaia di lire, potete costruirvi un sistema biprocessore basato sul processore Intel Celeron, e godere dei vantaggi di un sistema operativo che diventerà ancor più veloce e ancora più pronto in risposta ai vostri input.
BeOS utilizza dei **moduli esterni** simili a plug-in per gestire tutti i filetypes. Questo significa che dopo aver scaricato il supporto per il formato PNG ad esempio, qualsiasi applicazione è **virtualmente** in grado ora di supportare quel tipo di formato grafico. Aggiornarsi all'ultima versione di un formato, diventa ora una operazione semplicissima e amplifica le possibilità di utilizzo dell'intero sistema.
Ho avuto modo di provare il BeOS connesso ad una **rete mista** di PC e Macintosh. Sono stato in grado di vedere i volumi e le stampanti Appletalk (e di stamparci), e anche di riuscire a vedere le risorse condivise della rete Microsoft.
Al contrario, una volta attivato il servizio telnet e di ftp, la macchina BeOS è divenuta visibile all'interno del network. Un login via telnet da un PC con Windows, mi ha dato il controllo remoto pressochè totale della macchina BeOS, cosa che non è possibile nemmeno con WindowsNT, se non equipaggiato con software di terze parti. Tutto questo, grazie alla stretta parentela che il BeOS ha con il mondo dello Unix.
Copiare files, passare da una applicazione ad un'altra, fare ricerche sul disco... in generale il sistema è sempre velocissimo nella risposta e non presenta mai quei blocchi o quelle attese che abbiamo oramai imparato a conoscere, e a dare per scontate, in Windows. A parole, non ci si crede.
Il BeOS costa circa 90.000 Lire. Nella confezione troverete il CD dell'OS, svariate demo di programmi shareware e non, e un manuale che vi permetterà di muovere i primi passi nella vostra nuova casa. Eccellente è anche la documentazione disponibile in linea, in formato HTML come è oramai d'obbligo. Dovrebbe presto esserci un importatore ufficiale italiano, ma sembra che per ora per usufruire del supporto tecnico ci si debba servire dell'assistenza online dal sito della Be. Devo dire che in una occasione in cui ho effettivamente avuto bisogno di farne uso, ho ricevuto risposta dettagliata nel giro di due giorni. Speriamo che la Be sia in grado di mantenere sempre a questi livelli la qualità del servizio offerto.



HOME | SITE MAP | SEARCH | CONTACT | BE EUROPE

Be. in your senses

WELCOME TO BE

ABOUT BE, INC.
PRODUCT
BE STORE
SOFTWARE FOR BEOS
THE WORLD OF BE
SUPPORT
JOBS

BeOS: The Broadband Digital Media Operating System

Dell Offers BeOS at Gigabuys.com
Shogo: mobile armor division for BeOS
iDOT.com Ships BeOS Systems
BeDC Europe: Last Chance to Register
Download BeOS 4.5.2 Update

iDOT.computers

Release 4.5 is Here!

The Be Operating System (BeOS) is the only operating system designed from the ground up for the broadband digital media age.

buy BeOS now

If you're new to BeOS, explore our website to learn more about BeOS and Be, Inc. If you're already taking advantage of this revolutionary technology, get the latest BeOS software and join in the BeOS community.

Japanese Page

DEVELOPERS | PARTNERS | PRESS

SEARCH

Quale futuro?

La Be ha da poco seguito le orme della RedHat e ha pubblicamente offerto parte della sua società sul **mercato azionario**. Ci si aspettano grosse novità nei prossimi mesi, sia per quanto riguarda il software disponibile, sia per due nuove versioni pianificate una per il futuro prossimo, e una per qualche momento verso la fine dell'anno 2000. La nuova versione dovrebbe offrire, tra l'altro, il supporto **multiutente**, una maggiore quantità di drivers per il supporto hardware, e una serie di innovazioni non ancora confermate dalla casa madre.
BeOS, tramite di moduli, permette di leggere in maniera trasparente partizioni **FAT**, **FAT32**, **HPFS** (mac), **NTFS** e **Linux** (le ultime due con prodotti di terzi). Scambiare files fra i due mondi, non è mai stato così semplice. Un altro settore che sembrerebbe strizzare l'occhio al BeOS è quello dei **videogames**. Alcuni giochi (tra cui SimCity3000) sono stati annunciati per BeOS, mentre altri potrebbero essere prodotti non appena verrà assicurato il supporto per le maggiori schede 3D presenti sul mercato. Va detto che la TnT Riva è già supportata, mentre sono disponibili dei drivers in versione beta per le scheda della 3dFx. Da sempre, la disponibilità di videogiochi ha aiutato l'espansione di un determinato sistema operativo. Al giorno d'oggi, Windows è diventata la piattaforma principe per quanto riguarda l'intrattenimento ludico, superando in qualità e disponibilità prodotti analoghi disponibili per le console.

Conclusioni

Se avete sperimentato Linux, **probabilmente** vi **innamorerete** anche del BeOS. Se avete fino ad ora utilizzato solo software Microsoft, dopo aver provato il BeOS non vorrete più tornare sui vostri passi.
BeOS ha davanti un futuro **promettente**, ma non **predeterminato**. Molto dipenderà dalla politica commerciale della Be e dalla sua capacità di mantenere alto il livello di interesse come è stato fatto fino ad ora, contemporaneamente alla presentazione di **pacchetti software professionali** che siano finalmente in grado di dimostrare le potenzialità di questo gioiello informatico.
Se utilizzate il PC per lavoro, o solo per un utilizzo da ufficio, o non siete amanti del rischio derivante dall'installazione di software che non conoscete, allora probabilmente allo stato attuale delle cose il BeOS non è ancora adatto per voi.
Se invece fate parte di quella schiera di utenti che ama provare software nuovo, ha almeno tre sistemi operativi installati, e riesce a vedere i siti web anche dal terminale, allora il BeOS potrebbe offrirvi qualche sorpresa interessante.
Non resta che **attendere** i prossimi mesi, per capire se il BeOS sarà effettivamente "**il lato buono della forza**", oppure se sarà solamente un'altra **testa** caduta sotto ai piedi **dell'instancabile avanzata** di Windows e del suo cavaliere Bill Gates.



HOME | SITE MAP | SEARCH | CONTACT | BE, INC

Be. in your senses

WELCOME TO BE EUROPE

BeOS: The Broadband Digital Media Operating System

Be delivers BeOS, the operating system designed from the ground up for the broadband digital media age.

buy BeOS now

Using advanced design techniques BeOS provides an application environment that enables multistream video, audio, and graphics, simultaneously and reliably on low-cost hardware.

Release 4.5 is Here!

Applications

BeOS RESELLERS
DEVELOPERS | CUSTOMERS
PARTNERS | PRESS

What's new on Be Europe today?

Be Developer Conference Frankfurt, Germany, October 7- 8

This fall we will be hosting a Be Developer Conference. Developers wishing to attend should be versed in the basics of programming BeOS. The conference will be held October 7th and 8th at the Crowne Plaza Holiday Inn in Frankfurt, Germany. More details can be found here.

You can register online now!

PC World stocks BeOS!

PC World, the biggest computer chain store in the United Kingdom, now stocks BeOS! If you live in the UK, you will be able to find a shop near you, among their 77 resellers!

PC WORLD THE COMPUTER SUPERSTORE

Fujitsu Ships the Silverline with BeOS!

Fujitsu Computer, Germany, delivers the Silverline: the first BeOS compatible systems in Europe!

FUJITSU

BeOS R4.5 Update 2

Silvia Sammarro - carina - s.sammarro@oltrelinux.com

PLUTO MEETING NOI C'ERAVAMO

Padova 24-25 settembre

Nei giorni 24 e 25 settembre si è tenuto a Padova, sede del Pluto Linux User Group (il maggiore LUG italiano), l'annuale Pluto Meeting. L'evento è stato organizzato presso il Dipartimento di Elettronica ed Informatica dell' Università di Padova,che si è offerto di ospitare nelle sue aule stand e relatori. L' ambiente universitario padovano si è ormai rivelato ricco di fermenti e fortemente teso alla realizzazione di iniziative di gruppo: proprio qui infatti è stato costituito, dalla libera organizzazione dei docenti della facoltà, il gruppo di utenti Linux Pluto Lug, che è poi sfociato nella costituzione del Pluto quale organismo nazionale.

L' organizzazione delle due giornate era suddivisa per bacino di riferimento: la prima giornata, strutturata per muovere prevalente l' interesse di operatori e aziende, nella seconda invece è stata creata una maggiore apertura al "semplice" utente Linux. Il meeting ha suscitato il giusto grado di attenzione grazie anche alla presenza nella prima giornata di Andrea Arcangeli, sviluppatore di kernel, al punto da richiamare persino l'interessamento di Linux Today e Linux Weekly, che ne hanno riportato notizia tra gli avvenimenti del mese di settembre. Su web erano da tempo reperibili tutte le informazioni, e le istruzioni per la preiscrizione.

Ecco un piccolo resoconto di alcuni tra gli interessanti "talk" tenuti dai relatori, di cui era fitto il programma del Pluto Meeting. Il già nominato Andrea Arcangeli ha presentato una relazione sul kernel di Linux (tra codice SMP e "spinlocks"). Fabio Cazzin, amministratore di rete ha voluto condividere, al di là di ogni facile polemica tra software open source e proprietario, l'esperienza all'interno di un'azienda, della sostituzione di un server di rete Microsoft NT con Linux. Michele Dalla Silvestra e Antonio Gallo hanno parlato di distribuzioni, il primo della Debian/GNU Linux, il secondo invece della distribuzione Linux italiana, Bad Penguin. Una piccola parentesi sulla cultura hacker a cura di Alessandro Tiberti, lungo tutto il percorso BBS-Internet. Infine, esempio di rete "da ufficio" e problemi legati alla sicurezza, illustrati da Mauro Tedesco a riguardo di "Samba e la sua evoluzione".

La partecipazione al meeting è stata, come di consueto nello spirito Pluto, gratuita e libera, con pari attenzione rivolta a pubblico e sponsor. Tra gli espositori, una soddisfacente rappresentanza di tutto ciò che significa software "open source" oggi, negli stand che facevano da catalizzatori dell'interesse generale in alternanza alle relazioni presentate. La Debian (distribuzione che sposa in pieno la filosofia GNU del freesoftware) si è distinta questa volta per una "particolare", simpatica distribuzione: magliette con logo che sono velocemente andate a ruba, e che tutti indossavano come segno distintivo della partecipazione al meeting. (Ma altri desiderabili gadgets hanno entusiasmato i partecipanti al meeting: prevedibilmente, pinguini di varie dimensioni, in versione adesiva e anche nella inedita-almeno per qualcuno-versione di stoffa, cucibile. Da sabato circolano parecchie macchine, motorini, e boxer "happy Linux user" in più J). Una particolare segnalazione merita Prosa, "Progettazione e Sviluppo Aperto", societa' padovana che sviluppa per statuto solo freesoftware; e poi ancora RedHat, la famosa azienda autrice di una delle piu' diffuse distribuzioni, e infine I3 Icube (prodotti e servizi per il networking) e Compaq.

Un momento particolarmente interessante è stato, nel pomeriggio della seconda giornata, la presentazione dei gruppi LUG con l'intervento delle delegazioni di Firenze, Brescia, Verona ed altre, compresa la neonata comunita' genovese. L'incontro, centrato sul resoconto delle modalità di formazione di ognuna della delegazioni LUG intervenute, attraverso l'esperienza maturata sul campo, ha incontrato l'interesse di quanti, non avendo vicina alcuna comunità LUG , cercavano informazioni, spunti e consigli per cominciare a muoversi nella realtà locale. Ogni delegazione ha presentato, attraverso la mappa del LUG italiani (www.linux.it/LUG) il proprio sito, raccontando gli inizi, illustrando i progetti compiuti e presentando particolari iniziative.

La maggior parte del gruppi LUG infatti nasce come momento aggregativo, e solo in seguito, un po' per il necessario adeguamento tecnico dei soci, molto per la difficoltà iniziale a reperire attrezzature, locali, server ospiti, si passa alla definizione di progetti veri e propri Su tutte, particolarmente divertente la testimonianza del portavoce del LUG Brescia, dove si sta speriementando tra le altre cose, un'iniziativa interessante. L' istituzione di "Linux days", incontri periodici indirizzati principalmente ai principianti, per promuovere la conoscenza di Linux anche a basso livello.

Su tutto, un bellissimo clima di affiatamento, nonché l'immancabile interesse per gli argomenti proposti hanno reso possibile un reale e proficuo scambio di nozioni tecniche e conoscenze personali durante un appuntamento che è, insieme, aggregazione e trasmissione, fotografia della situazione Linux italiana e occasione per azzerare almeno annualmente le distanze reali.
Il pomeriggio è sfumato sui numerosi, movimentati gruppetti che discutevano animatamente l'argomento principe "non ufficiale": meglio slackware o debian ? Aspettiamo il Meeting Y2K.



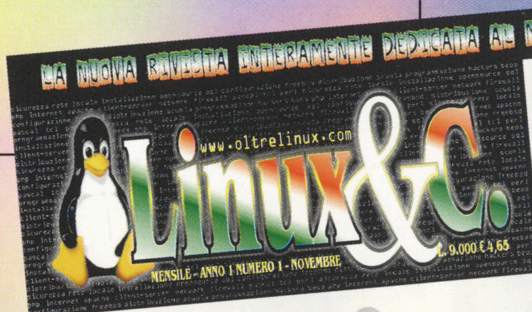
Che altro dire se non...

Mousepad in omaggio con questo numero



Questo mese doppia rivista: anche il numero 0 nel cd in formato PDF.

Recensioni: solo i migliori tools e softwares su Linux&C.



DI QUESTO NUMERO
- WORD PERFECT 8
- INTERNET
- DOCUMENTAZIONE
- MASTERIZZAZIONE
- SVILUPPO

RECENSIONE
Corel WordPerfect 8
Un facile e potente WordProcessor, perchè non si pensi che Linux sia sprovvisto di strumenti di produttività. Ed è solo l'inizio...

HACKERS-MANIA
Difendiamoci con i firewall
I firewall sono lo strumento di sicurezza più utilizzato in assoluto. Cerchiamo di capire cosa sono, come si installano e come si configurano.

CONFIGURAZIONE
INTERNET facile
Configurare l'accesso ad internet può presentare qualche difficoltà. Una guida per i nuovi utenti.

ED ANCORA...
- INSECURITY NEWS
- INSTALLAZIONE
- PROGRAMMAZIONE

LINUX A SCUOLA
NetQuiz: un progetto «scolastico»
Continua lo sviluppo del software per la realizzazione di Quiz. Un progetto client-server su piattaforma mista Linux/Windows.

BeOS: un nuovo sistema operativo dalle caratteristiche interessanti

Sicurezza: una rubrica fissa nelle pagine del magazine.

Un Cd-Rom interamente dedicato al mondo Linux.

Programmazione: C/C++ specifico per Linux, Java, XML, FreePascal e molto altro...

Configurazione: tutti i principali aspetti di Linux verranno trattati ed ampiamente discussi.



Linux&Scuola: una sezione dedicata allo sviluppo di soluzioni per la scuola. Da seguire assolutamente!

Nel prossimo numero

- Prove: Corel Linux.
- Sicurezza: firewall 2ª parte.
- Multimedia: tutto sugli Mp3.
- NetQuiz: Linux/Windows.
- Internet: configurazione posta.
- Apache+PHP+SQL: ultima puntata.
- DNS: introduzione.
- File Server: SAMBA.
- XML/Java
- Programmazione: un nostro simpatico progetto opensource in freepascal e C++.

...che Vi aspettiamo in edicola il 15 novembre?